

Software Engineering Department

Braude College

**Predicting Life Expectancy Based On Personal, Lifestyle, and Medical Indicators Using Machine Learning**

**[25-1-R-12]**

By:

Maher Salman

Adan Butto

Advisors :

Dr. Naomi Unkelos-Shpigel

Dr. Samah Idrees Ghazawi

Website : <https://life-expectancy-pi.vercel.app/>

GitHub: <https://github.com/mahersalman/life-expectancy>

**Table of Contents**

[**Abstract 3**](#_1fob9te)

[**1. Introduction 3**](#_30j0zll)

[**2. Related Work - Literature Review 4**](#_gbuhcfdtd86s)

[2.1. Life Expectancy Research: Key Studies and Predictive Models 5](#_e9iumx3j34ox)

[**2.2. Key Indicators for Life Expectancy 6**](#_3znysh7)

[2.3. Machine learning in life expectancy 7](#_x4mqd13nrtv9)

[**3. Methodology and Research Process 8**](#_efsmo7b90bg7)

[3.1. Data Source And Penalty 8](#_jkegvyeqaqsd)

[3.1.1. Personal Key Indicators of Heart Disease Dataset [25] 8](#_dqfzxt70fhk3)

[3.1.2. Life Expectancy and Healthy Life Expectancy Dataset [34] 9](#_l1ssio7kzkjz)

[3.1.3. Applied Penalty Adjustments 9](#_7hm5kf4vbon)

[3.2. Preprocessing 10](#_7fxrs9oap5ul)

[3.2.1. Synthetic Life Expectancy Labeling Process 10](#_u87dr6lxe0sx)

[3.2.2. Analysis of the Generated Dataset 11](#_yxgghi75ote6)

[3.3 Predictive Modeling Approaches for LE 17](#_qhdx9u276gla)

[3.3.1 XGBoost 18](#_wu1t0xr0qoe8)

[3.3.1.1 Flow of the XGBoost Model 19](#_wh4oqw23oehx)

[3.3.1.2 Example: Step-by-Step Flow of the XGBoost Model 21](#_jv4kd3a07sdw)

[**4. Evaluation 22**](#_fomguh5gn5ub)

[4.1. Metrics in Detail 23](#_eq2wuqv7g6kp)

[**5. Project Architecture and Deployment 24**](#_4iw9ifh7az0f)

[**6. Results 25**](#_twaxqh43flvw)

[**7. User Guide 28**](#_uxc4ps4iaa2n)

[7.1 Key Features That Enhance User Experience 28](#_3mfzc2kaxwi8)

[7.2. Explanation with screenshots 29](#_etbdh1sxsnkx)

[**8. Maintenance Guide 41**](#_vab66v1dt2hg)

[8.1 Environment 41](#_l4619iqqq16d)

[8.2 Installation Instructions: 41](#_atu55kqhpzhp)

[8.3 Synthetic Data Generation & XGBoost Training 42](#_epehljssdqyr)

[8.4 Life Expectancy Next.js Website (Front-End) 43](#_2xo01exvgj77)

[8.4.1 Entry Point (src/app/page.tsx) 43](#_rh9netwdfzcs)

[8.4.2 Core Components 44](#_jeoe61ntq5x8)

[8.4.3 Context Providers 45](#_msj3p7mwcwf0)

[8.4.4 Utility Modules 45](#_yyn8exwlz1am)

[8.4.5 Translation and Language Support 45](#_gxyag5492k9d)

[**9. Usability Evaluation and Feedback Insights 46**](#_64d00kfzqut)

[**10.Documenting Failed Experiences During the Study 48**](#_ucotqwyy0zst)

[10.1. Synthetic Dataset Creation Using Averaged Data 48](#_qzj4t8d5wn70)

[10.2. Challenges Encountered in Deep Learning Approach 49](#_uq11pxate2wr)

[**References: 51**](#_3dy6vkm)

# Abstract

Life expectancy (LE) is a fundamental indicator of public health and quality of life. Despite its significance, there is a lack of publicly available, person-level datasets suitable for predictive modeling of life expectancy based on individual characteristics. To address this gap, we constructed a fully synthetic, individual-level dataset informed by real-world data from reputable sources, including the World Health Organization (WHO) and the Behavioral Risk Factor Surveillance System (BRFSS), conducted by the Centers for Disease Control and Prevention (CDC). Both source datasets were obtained from Kaggle. The final dataset includes 26 features grouped into four key domains: personal information (e.g., sex, BMI), lifestyle behaviors (e.g., smoking, physical activity), medical history, and preventive care (e.g., vaccination status). The effect of each individual indicator on life expectancy is analyzed in Sections 2.2 and 3.1.3, highlighting their relative influence on lifespan.

Using this dataset, we trained and evaluated an XGBoost regression model that achieved strong predictive performance, with an RMSE of 5.73 years, MAE of 4.4 years, and R² of 0.65—indicating that 65% of the variability in life expectancy across individuals is explained by the model. Residual analysis confirmed that the model is both unbiased and stable across prediction ranges. These results demonstrate that machine learning models, when trained on high-quality synthetic data guided by real-world patterns, can deliver accurate, interpretable life expectancy predictions and support public health decision-making and personalized care strategies.

# 1. Introduction

Life expectancy (LE) is vital to public health and societal well-being. It offers insights into the average potential longevity of individuals within a population. For individuals at a given age, it estimates the additional years they are likely to live based on prevailing death rates. By summarizing mortality patterns, life expectancy serves as a key indicator of health outcomes, societal progress, and the effectiveness of healthcare systems [1].

Predicting life expectancy is a multifaceted challenge influenced by a wide range of factors. Personal attributes such as body mass index (BMI), physical activity levels, and smoking habits interact with systemic determinants like access to healthcare, socioeconomic status, and environmental conditions. These variables often have complex, non-linear relationships, making life expectancy prediction both complicated and rich in potential insights. Accurate predictions of LE are essential for public health planning, enabling policymakers to identify key determinants and design targeted interventions to improve health and longevity [6].

While previous research has explored individual indicators of life expectancy, integrating machine learning presents a transformative approach to tackling this problem. Machine learning algorithms allow for the simultaneous analysis of multiple factors, uncovering intricate patterns and relationships with exceptional precision and scalability [1]. By leveraging these capabilities, predictive models can be constructed to estimate life expectancy with greater accuracy than traditional statistical methods.

This project builds on recent advancements by utilizing real-world data from diverse sources to ensure a comprehensive representation of factors influencing life expectancy. These factors are grouped into four main categories: personal information, lifestyle behaviors, medical history, and preventive care. The prediction model is based on XGBoost, a state-of-the-art machine learning algorithm widely recognized for its effectiveness on tabular data [7]. Through careful model optimization and validation, the results are designed to be both accurate and generalizable, providing valuable tools for improving public health outcomes across diverse populations.

The main outcome of this project is the development of a web-based life expectancy calculator powered by a machine learning model trained on a realistic, synthetic dataset. Although the dataset is artificially generated, it is grounded in real-world distributions and relationships, allowing the model to accurately simulate the impact of various personal and health-related factors on life expectancy. The system provides users with an interactive way to explore how changes in specific indicators—such as smoking habits, physical activity, or preventive care behaviors—can increase or reduce their predicted lifespan. The model handles these changes effectively, clearly illustrating the directional impact of each factor. This result highlights the tool’s potential as an educational and awareness-raising platform for the general public, enabling individuals to better understand the consequences of lifestyle and medical decisions on longevity.

The remainder of this report is organized as follows: Section 2 reviews related work in life expectancy modeling and presents the set of indicators used in this study. Section 3 outlines the methodology and research process, including the creation of the synthetic dataset using a penalty-reward framework and the application of the XGBoost algorithm for prediction. Section 4 describes the evaluation strategy and performance metrics used to validate the model. Section 5 details the system architecture, including both backend implementation of the machine learning model and the frontend user interface. Section 6 presents the experimental results and model performance. Section 7 serves as a user guide, explaining how to use the website. Section 8 provides a maintenance guide for future developers. Section 9 discusses the usability evaluation and feedback gathered from other students and users. Finally, Section 10 reflects on alternative ideas explored during the project, including unsuccessful attempts in generating the synthetic dataset and applying deep learning techniques to tabular data.

# 2. Related Work - Literature Review

Understanding the factors influencing life expectancy is a critical area of research with profound implications for public health and policy-making. Numerous studies have explored the relationship between health, economic, and social indicators and life expectancy, leveraging both traditional statistical approaches and modern machine learning techniques. These works provide valuable insights into the predictors of longevity and form the foundation for more advanced modeling and analysis.

**However, a common limitation across most existing studies is their reliance on country-level average data**, typically sourced from international organizations like the World Health Organization (WHO). While useful for global or national trend analysis, these aggregated datasets lack the granularity required to build accurate models for predicting life expectancy at the **individual level**. This gap highlights the need for synthetic or real person-level datasets that can support personalized life expectancy modeling, which is the focus of our approach.

## 2.1. Life Expectancy Research: Key Studies and Predictive Models

1) The article **“An Application of a Supervised Machine Learning Model for Predicting Life Expectancy”** investigates the use of machine learning techniques, specifically the Extreme Gradient Boosting (XGBoost) algorithm, to forecast life expectancy across various countries. The study focuses on identifying key determinants influencing life expectancy by analyzing data from the World Health Organization (WHO) and the United Nations (UN).

Key factors examined include:

**Health Indicators**: Adult mortality rate, infant deaths, immunization coverage (e.g., Polio, Hepatitis B), and disease prevalence (e.g., HIV/AIDS).

**Economic Factors**: GDP per capita, total health expenditure, and alcohol consumption.

**Social Factors**: Average years of schooling, population demographics, and country development status.

The XGBoost algorithm demonstrated exceptional performance, surpassing earlier models in accuracy and handling dataset complexity. The results highlighted significant predictors of life expectancy, such as adult mortality rate, GDP per capita, and immunization coverage. These findings emphasize the importance of enhancing healthcare access, economic growth, and disease prevention to positively impact life expectancy [1].

**2)** Das, Uddin, and Karim (2025) used WHO average data per country to compare eight machine learning models—including Linear Regression, Decision Tree, Random Forest, SVM, KNN, Gradient Boosting, XGBoost, and Neural Networks—for predicting life expectancy across countries. They applied Boruta and Regularized Random Forest feature selection to identify the top 20 predictors, which included socioeconomic, health, and demographic variables. Among the models, **Random Forest achieved the best performance** with **R² = 0.969**, **RMSE = 0.179**, and **MAE = 0.116**. XGBoost also showed strong performance with **R² ≈ 0.96**, and they conclude that **Random Forest, XGBoost, and Gradient Boosting are the most accurate models for predicting life expectancy** [5].

**3)** In the systematic review *"****Machine Learning Algorithms in Personalized Life Expectancy Prediction: A Systematic Review****"*, the author examines machine learning approaches applied to individual-level life expectancy prediction using person-specific data ranging from demographic and clinical attributes to electronic health records. The review focuses on three primary algorithms: Random Forest, XGBoost, and Neural Networks, analyzing their performance across various healthcare datasets. XGBoost demonstrates superior performance compared to Random Forest because it boosts the performance of trees by constructing them step by step and using advanced regularization. Overall, it gives better results than RF when the situation is not very noisy, and the data is not too complex. Due to its guard against overfitting, it is an excellent choice for programs working with real clinical data that is sometimes unreliable and poor in quality. The review identifies that ensemble methods consistently show deep learning is often the most accurate, followed fairly closely by XGBoost and then RF , though implementation challenges include limited transparency in model interpretability and the need for careful validation protocols to ensure equitable outcomes across diverse patient populations [2].

## 2.2. Key Indicators for Life Expectancy

Drawing on established research, we have identified a definitive set of indicators—each shown to correlate strongly with life expectancy—as the sole basis for our analysis. Employing machine learning techniques, we will assess the predictive strength of these chosen factors and quantify their individual contributions. Below, we present each indicator, summarize our findings, and discuss its relevance to life expectancy modeling.

**Table 1. PersonaInfo**

|  | Indicator | Explanation |
| --- | --- | --- |
| 1 | Body Mass Index (BMI) | Measure of body fat based on weight and height; used to categorize health risks. |
| 2 | sex | Biological sex affecting health risks and life expectancy differences. |

**Table 2. Life Style**

|  | Indicator | Explanation |
| --- | --- | --- |
| 1 | Smoking and Vaping | Usage of electronic cigarettes and tobacco, significantly impacting both respiratory and cardiovascular health. |
| 3 | Physical Activities | regular physical exercise, impacting overall health and longevity. |
| 4 | Sleep Hours | Average nightly sleep duration, influencing physical and mental health. |
| 5 | Alcohol Consumption | Frequency and quantity of alcohol intake, affecting liver and overall health. |

**Table 3. Medical History**

|  | Indicator | Explanation |
| --- | --- | --- |
| 1 | Heart Attack | History of myocardial infarction, critical for cardiovascular risk assessment. |
| 2 | Angina | History of chest pain due to reduced blood flow to the heart. |
| 3 | Stroke | Past incidents of cerebrovascular accidents affecting brain function. |
| 4 | Asthma | Presence of chronic respiratory disease affecting breathing. |
| 5 | COPD | Chronic obstructive pulmonary disease, severely affecting respiratory health. |
| 6 | Depressive Disorder | History of clinical depression impacting mental and physical health. |
| 7 | Kidney Disease | Chronic kidney disease history, significant for renal function assessment. |
| 8 | Arthritis | Chronic joint inflammation affecting mobility and quality of life. |
| 9 | Diabetes | Presence of diabetes mellitus, impacting metabolic and cardiovascular health. |
| 10 | deaf or hard of hearing | Untreated hearing loss has also been associated with increased rates of anxiety, depression, poor mental health and lower life expectancy. |
| 11 | blind or vision difficulty | the risk of mortality was 29% higher for participants with mild vision impairment, compared to normal vision. The risk increases to 89% among those with severe vision impairment. |
| 12 | difficulty concentrating | The researchers conclude that, on average, people with an ADHD diagnosis have shorter lives than those without the condition. |
| 13 | difficulty walking | Mobility impairment significantly affecting independence and health. |
| 14 | difficulty dressing bathing | Difficulty in performing personal care activities, indicating severe functional limitation. |
| 15 | difficulty errands | Difficulty in completing routine tasks outside home, affecting independence. |

**Table 4. Preventive Care**

|  | Indicator | Explanation |
| --- | --- | --- |
| 1 | Influenza Vaccine (Past Year) | Receipt of influenza vaccination in the past year, relevant for infection prevention. |
| 2 | Pneumococcal Vaccine | History of receiving pneumococcal vaccination, important for pneumonia prevention. |
| 3 | Tetanus Vaccine (Past 10 Years) | Receipt of tetanus vaccine within last ten years, significant for tetanus prevention. |
| 4 | High-Risk Status (Within Last Year) | Identification as high-risk individual within the last year, indicating increased vulnerability to health issues. |

## 2.3. Machine learning in life expectancy

In related studies, commonly used machine learning techniques for life expectancy prediction include linear regression and multiple regression models, which are effective for analyzing relationships between indicators and life expectancy. Ensemble methods such as random forests and decision trees are widely employed for feature importance analysis and to enhance predictive performance [6]. Advanced techniques like support vector machines (SVM) are utilized to model complex non-linear relationships, while gradient boosting algorithms, such as XGBoost, are favored for handling large datasets and optimizing predictions [1].

Despite these advancements, existing approaches still face limitations. A key challenge lies in the generalizability of these models, as they often struggle to adapt to diverse datasets from different countries or regions [4]. Additionally, publicly available datasets tend to represent averaged data [31], which weakens the ability to identify precise relationships between life expectancy and specific indicators. Addressing these gaps is crucial for developing more accurate, adaptable, and practical life expectancy prediction models that can offer actionable insights to inform global health policies and strategies.

# 3. Methodology and Research Process

We began by surveying the landscape of life-expectancy research and assembling a set of strongly correlated indicators. In our search for granular, individual-level data, we found that only aggregate datasets—most notably from the World Health Organization (WHO)—were publicly available. To work around this limitation, we developed a penalty-based synthesis approach that integrates individual-level health data from the BRFSS dataset with country-level life expectancy statistics from the WHO.

Our method assigns a synthetic life expectancy value to each record by referencing the appropriate life-table average (e.g., life expectancy at birth or at age 60) and then adjusting it using evidence-based penalties and bonuses. These adjustments account for health-related factors such as smoking, alcohol use, chronic disease presence, BMI, physical activity, vaccination status, and sleep patterns. Controlled noise and bounds ensure the resulting synthetic labels remain realistic while capturing individual variation. This process results in a tabular dataset suitable for training machine learning models to predict life expectancy at the individual level.

With our synthetic tabular dataset in hand, we turned to models proven effective on structured medical data. We selected XGBoost—a gradient-boosting implementation renowned for top Kaggle performances and robust handling of heterogeneous features—and complemented it with a deep-learning architecture tailored to tabular inputs. Each model is rigorously optimized and evaluated on our dataset so we can quantify predictive accuracy, compare strengths and weaknesses, and determine which approach best forecasts life expectancy.

## 3.1. Data Source And Penalty

For this study, two datasets have been employed to enhance analysis accuracy and relevance. Both datasets were obtained from Kaggle, a popular platform for publicly shared data. The datasets are integrated such that individual health indicators from the heart disease dataset [25] are combined with country-level life expectancy benchmarks from the WHO dataset [34]. Subsequently, individual life expectancies are adjusted using penalty scores based on specific health indicators, as detailed in Table 5 of this research.

### 3.1.1. Personal Key Indicators of Heart Disease Dataset [25]

This dataset includes 2022 annual survey data from over 400,000 adults, provided by the Centers for Disease Control and Prevention (CDC). Originally part of the Behavioral Risk Factor Surveillance System (BRFSS), the dataset has been condensed from nearly 300 variables to the 40 most relevant indicators associated with heart disease, including blood pressure, cholesterol levels, smoking status, diabetes, obesity (BMI), physical activity, and alcohol consumption. The primary variable of interest, "HadHeartAttack," is treated as a binary classification indicating the occurrence of heart disease. Due to class imbalance, special consideration such as undersampling or adjusting model weights is recommended when applying machine learning methods. This dataset is utilized for individual-level health data analysis.

### 3.1.2. Life Expectancy and Healthy Life Expectancy Dataset [34]

Provided by the World Health Organization (WHO), this dataset offers comprehensive life expectancy metrics by country and wealth group, covering historical data from 2000 through 2019. It is used to calculate an average life expectancy baseline for each country, serving as a reference to synthetically determine individualized life expectancy for participants from the first dataset.

### 3.1.3. Applied Penalty Adjustments

To generate our synthetic cohort, we assign evidence-based year-penalties (or gains) for each risk factor—such as smoking, physical activity, sleep duration, and heart attack history—ensuring the simulated data reflect real-world life-expectancy impacts documented in the literature.

**Table 5. Penalty Adjustments for Synthetic Data**

|  | **indicator** | **explanation** | **REF** |
| --- | --- | --- | --- |
| 1 | Smoking and Vaping | Associated with a 6.8 to 8.8 year reduction in life expectancy. | [8] |
| 2 | Physical Activity | Regular activity can increase lifespan by 0.4 to 6.9 years. | [9] |
| 3 | Sleep Duration | Sleeping fewer than 7 hours or more than 9 hours is associated with a reduction of 1–3 years in lifespan. | [10] |
| 4 | Heart Attack | Results in an average reduction of 12 years in life expectancy. | [17] |
| 5 | Stroke | Decreases lifespan by 5.5 to 7.4 years. | [18] |
| 6 | Vaccinations (Influenza, Pneumococcal, Tetanus) | Vaccines have significantly increased global life expectancy, preventing at least 154 million deaths over the past 50 years and contributing to decades of added healthy life per person saved. | [23] |
| 7 | Body Mass Index (BMI) | * BMI ≥ 30: Reduces life expectancy by 4.2 years in men and 3.5 years in women. * BMI ≤ 18.5: Reduces life expectancy by 4 years | [19] |
| 8 | Asthma | Associated with a reduction in life expectancy by approximately 3.3 years. | [20] |
| 9 | Depression | Found to reduce life expectancy by 12 to 21 years, depending on severity and comorbidities. | [11] |
| 10 | Kidney Disease | Linked to a decrease in life expectancy by 6 years. | [12] |
| 11 | Rheumatoid Arthritis | May reduce life expectancy by up to 10 years. | [24] |
| 12 | Disabilities | Varying degrees of disability can lead to a reduction of up to 10 years in life expectancy. | [13] |
| 13 | Alcohol Consumption | Reduces life expectancy by approximately 0 to 6.9 years, depending on usage patterns. | [14] |
| 14 | Diabetes | Shown to decrease lifespan by around 6 years. | [15] |
| 15 | COPD (Chronic Obstructive Pulmonary Disease) | * Current smokers: 2.2–5.8 years reduction * Former smokers: 1.4–5.6 years reduction * Never smokers: 0.7–1.3 years reduction | [16] |

## 3.2. Preprocessing

Our preprocessing pipeline ingests two primary CSV files—heart\_2022\_with\_nans.csv (individual-level BRFSS records, many with missing values) and xmart.csv (country-year life-expectancy averages)—and transforms them into a unified, analysis-ready dataset. We begin by extracting official life-table statistics from the xmart.csv file to establish baseline life expectancy parameters by sex and age categories. The individual-level data undergoes comprehensive preprocessing including dropping irrelevant or redundant fields and mapping all categorical responses (e.g., "Yes/No," gender, smoking status, e-cigarette usage, tetanus vaccination) to consistent numeric codes. Missing values are handled through appropriate imputation methods, and extreme outliers are managed to reduce undue influence. A synthetic life expectancy target variable is then generated using a sophisticated algorithm that incorporates the official life-table statistics from xmart.csv as baseline parameters, combined with individual health factors, lifestyle behaviors, and disease conditions through realistic penalty and benefit calculations. All numerical features are standardized using z-score normalization to ensure comparable scaling across variables. Finally, the cleaned individual records are split into training (80%) and test (20%) sets using stratified sampling to preserve the original distributions of life expectancy and key demographics for robust downstream modeling.

### 3.2.1. Synthetic Life Expectancy Labeling Process

As part of our preprocessing workflow, we augment the cleaned BRFSS records with a fully synthetic life-expectancy label. From xmart.csv we extract four benchmark columns—LE\_birth\_male, LE\_birth\_female, LE\_60\_male, and LE\_60\_female—and compute their official minimum, maximum, and mean values. Each preprocessed individual record then receives a synthetic LifeExpectancy via a custom function that:

(1) selects the appropriate life-table mean (birth or age 60) plus modest Gaussian jitter.

(2) applies evidence-based penalties for smoking intensity, alcohol use, sleep deviations, chronic diseases, disabilities, and extreme BMI (capped for plausibility).

(3) adds benefits for physical activity and vaccinations.

(4) introduces final random noise.

(5) clamps the result within the official min/max bounds.

Embedding these steps in preprocessing ensures that our downstream ML models train on an individual-level dataset whose marginal and joint distributions faithfully mirror real-world demographics—addressing the critical absence of any public person-level life-expectancy dataset.

### 3.2.2. Analysis of the Generated Dataset

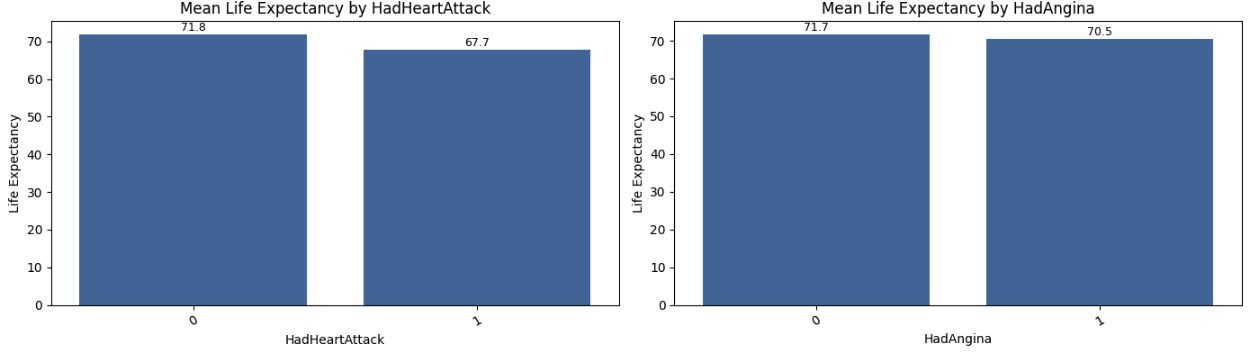
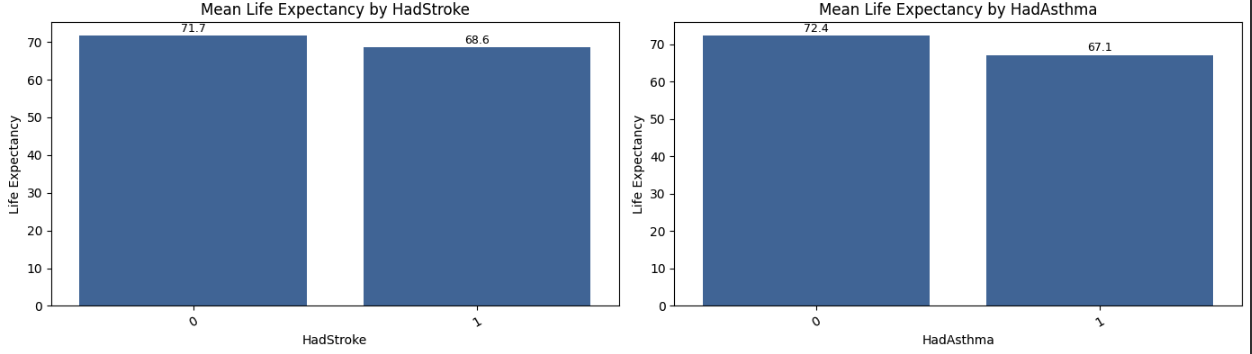
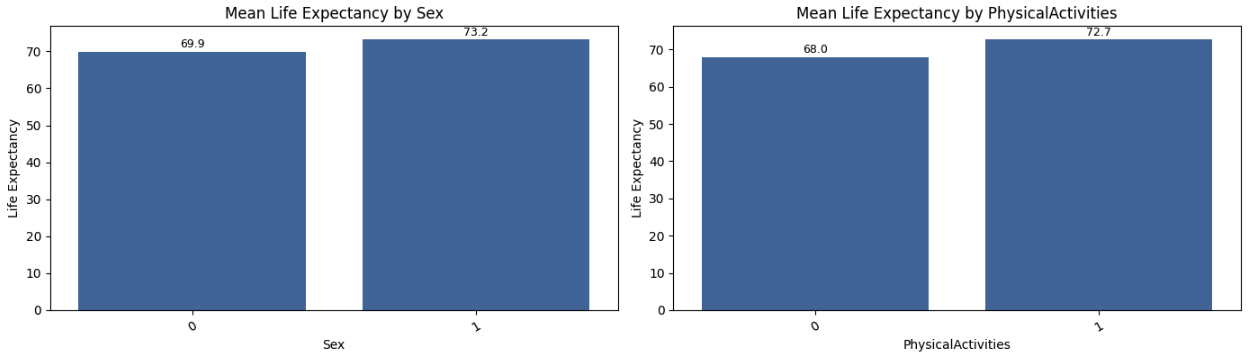
The synthetic cohort yields a lifespan distribution that is highly plausible for an adult population: over 50 percent of simulated individuals fall in the 70–79 age bracket, nearly 20 percent in the 80–89 range, and virtually none below age 40 (**Fig. 3**). Crucially, these post-penalty “adjusted” lifespans still mirror real-world demographics, demonstrating that our evidence-based penalties and gains did not introduce implausible artifacts.

Next, we quantified each indicator’s effect on these adjusted lifespans. A Pearson correlation analysis (**Fig. 1**) identifies depressive disorder (r = –0.52), difficulty concentrating (r = –0.31), and smoking status (r = –0.29) as the strongest negative predictors of life expectancy, whereas influenza vaccination (r = +0.21) and regular physical activity (r = +0.20) show the largest positive associations. Category-level bar plots in **(Figs. 2.1-2.6**) (participant characteristics: Sex, Physical Activities, Heart Attack, Angina, Stroke, Asthma), (**Figs. 2.7-2.12**) (health conditions: COPD, Depressive Disorder, Kidney Disease, Arthritis, Diabetes, Hearing Difficulty), (**Figs. 2.13 - 2.18**) (functional limitations & Smoking Status: Vision Difficulty, Concentration, Walking, Dressing/Bathing, Errands, Smoker Status), and (**Figs.2.19-2.24**) )(preventive behaviors: E-cigarette Usage, Alcohol Consumption, Influenza & Pneumococcal Vaccination, Tetanus Booster, High-Risk Condition) further illustrate—for example—that smokers lose roughly 6–9 years compared to never-smokers and that physically active individuals gain about 4–7 years over inactive peers, validating that our penalty adjustments faithfully reproduce documented epidemiological effects.

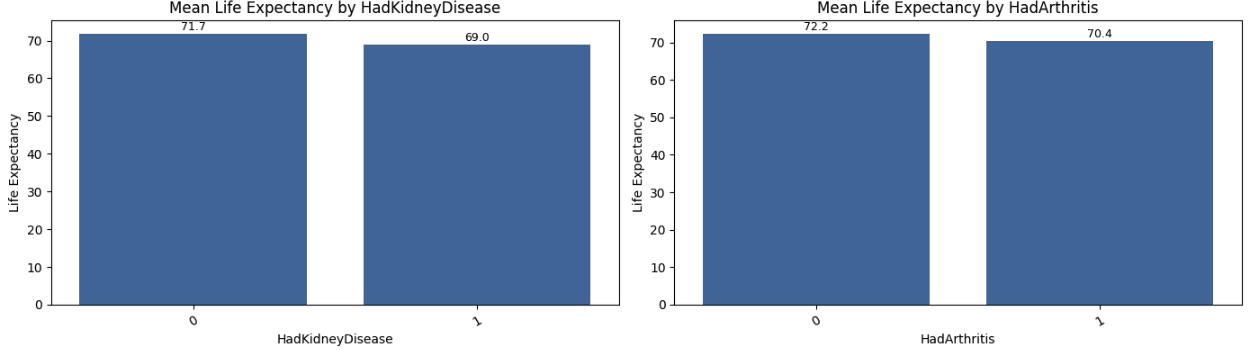
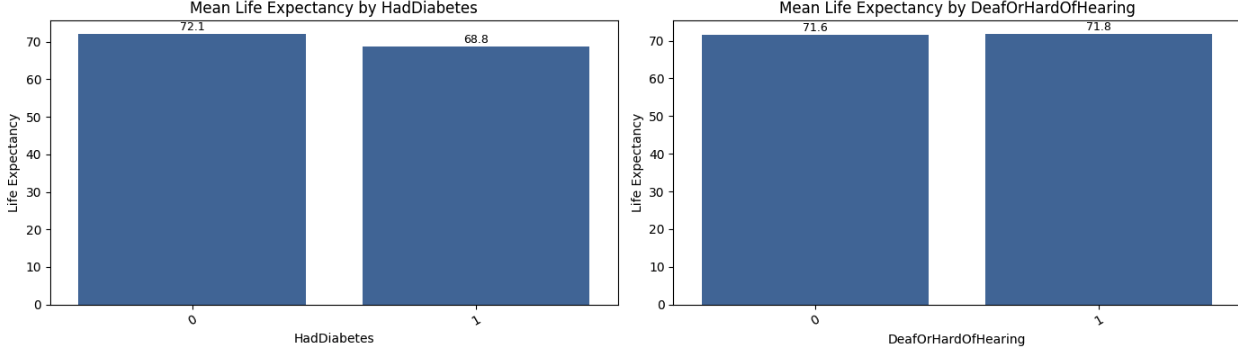
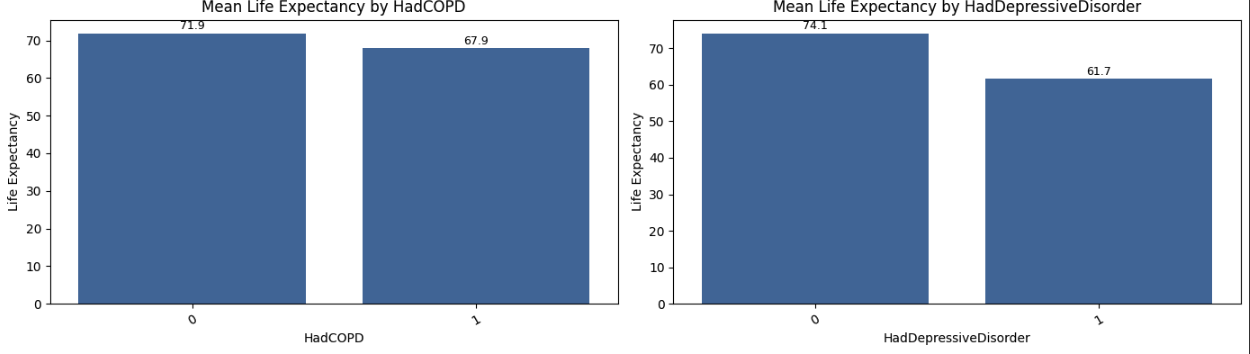
In the next section, we leverage these insights to inform feature selection and weighting in our predictive modeling pipeline. We then describe the design, training, and evaluation of an XGBoost regressor whose performance underscores the value of incorporating these empirically grounded penalties and gains.

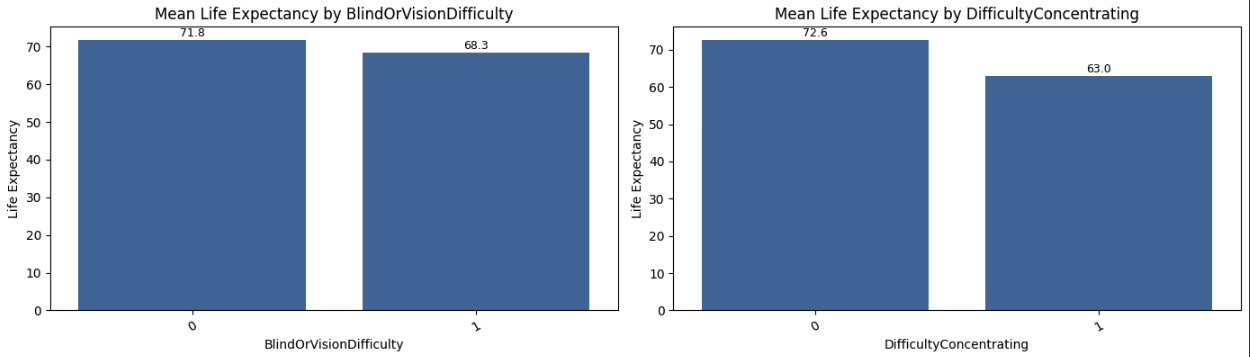
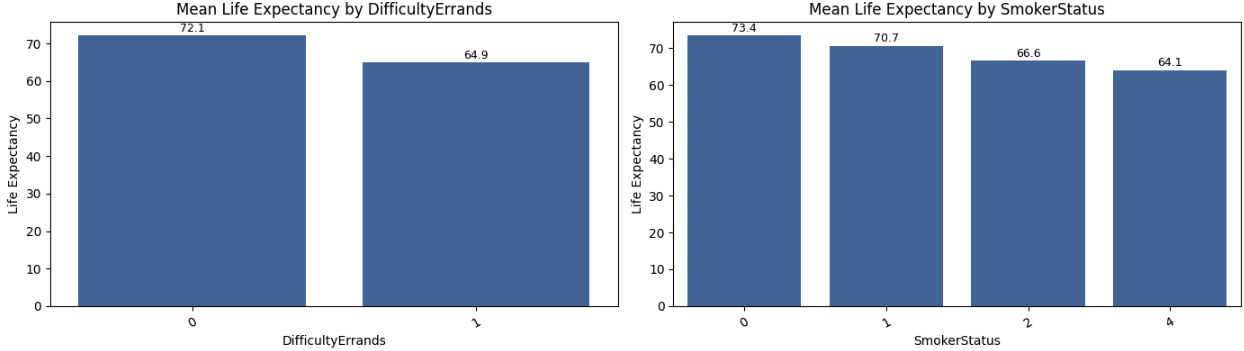
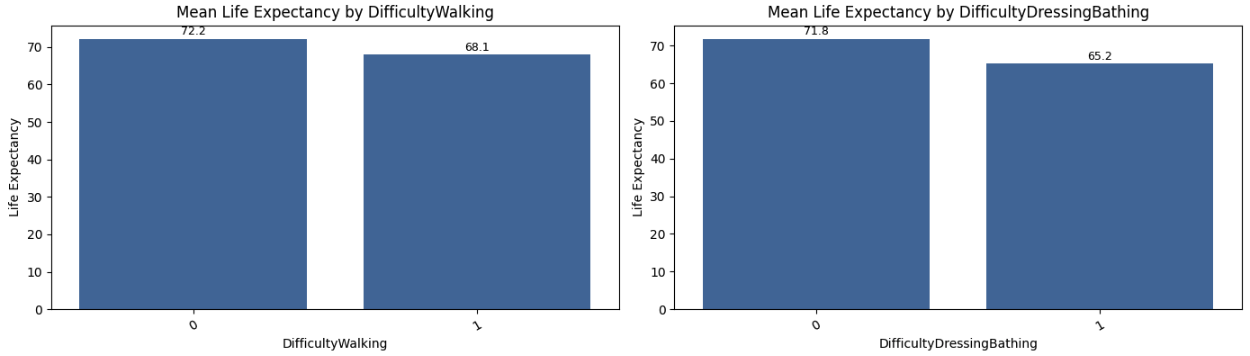
****

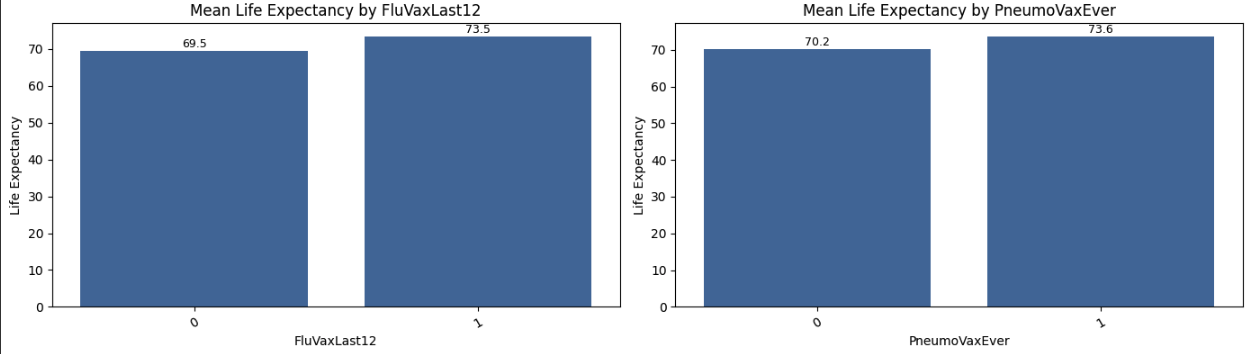
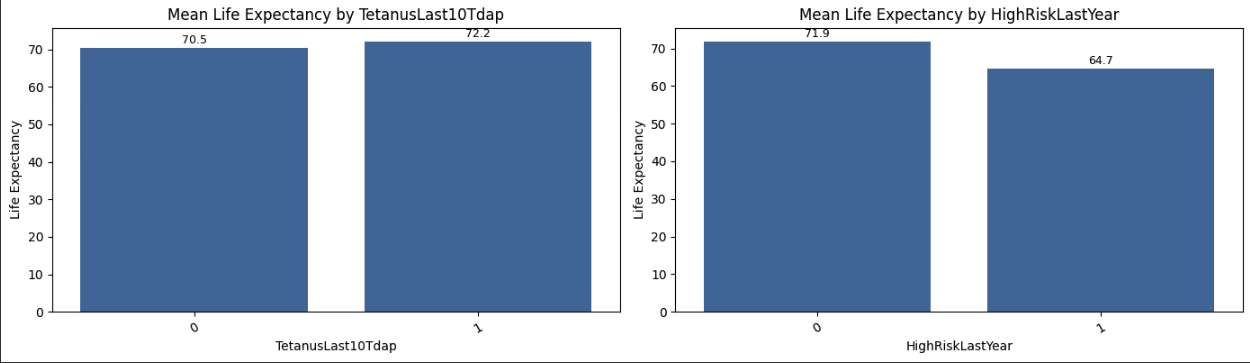
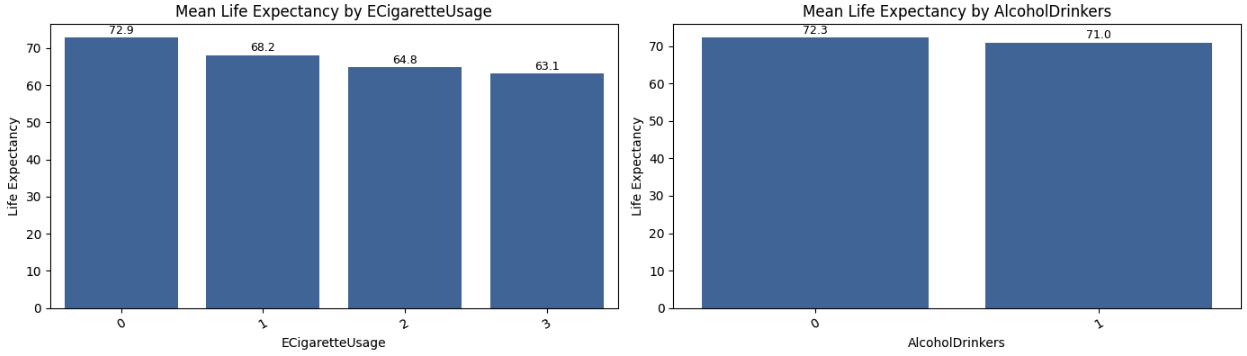
**Fig. 1 - Correlation of Each Indicator with Synthetic Life Expectancy**

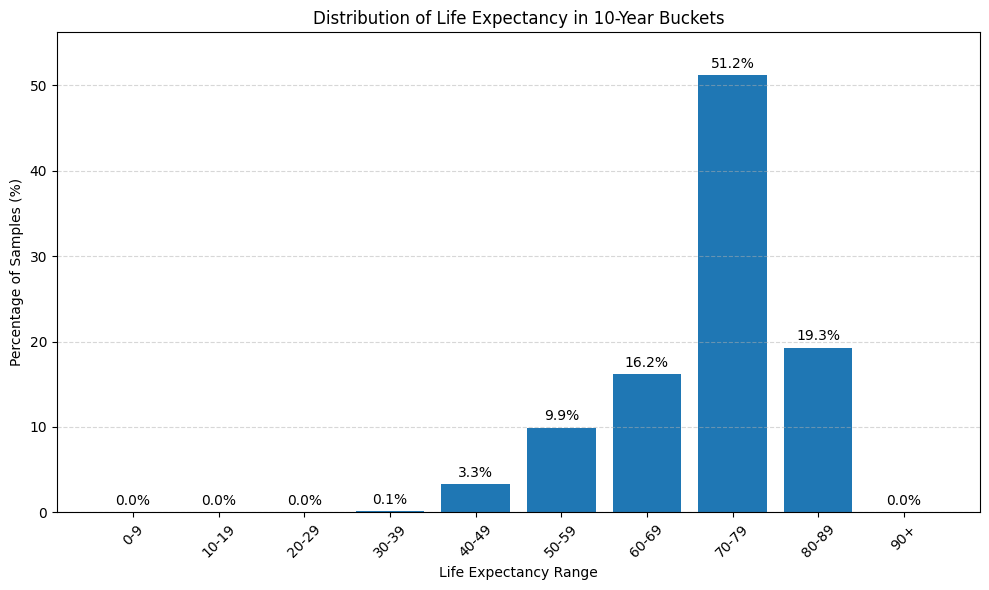


**Fig. 2.1 - 2.6 - Mean life expectancy by participant characteristics: Sex; Physical Activities; history of Heart Attack; history of Angina; history of Stroke; and history of Asthma.**

**Fig. 2.7- 2.12 - Mean life expectancy by participant health conditions: history of COPD; history of depressive disorder; history of kidney disease; history of arthritis; history of diabetes; and deafness or hearing difficulty.**

**Fig. 2.13-2.18 - Mean life expectancy by functional limitations and smoking status: blind or vision difficulty; difficulty concentrating; difficulty walking; difficulty dressing/bathing; difficulty running errands; and smoker status.**

**Fig. 2.19 - 2.24 - Mean life expectancy by e-cigarette usage; alcohol consumption; influenza vaccination in the past 12 months; pneumococcal vaccination ever; tetanus vaccination within the last 10 years; and high-risk health condition in the past year.**

**Fig. 3 - Distribution of Synthetic Life Expectancy in 10-Year Buckets**

## 3.3 Predictive Modeling Approaches for LE

We predict life expectancy using XGBoost, a gradient-boosting ensemble algorithm renowned for its speed, flexibility, and strong performance on structured data. XGBoost builds successive decision-tree learners that iteratively correct the errors of their predecessors, yielding highly accurate and robust models even when complex feature interactions and non-linearities are present. It has demonstrated significant effectiveness in medical applications, handling complex and diverse datasets while meeting accuracy requirements for auxiliary diagnosis [3].

For our implementation, we perform a grid search over key hyperparameters—number of trees, maximum tree depth, learning rate, and subsampling ratios—using five-fold cross-validation on the training set to guard against overfitting. Model performance is then evaluated on held-out validation and test splits using RMSE, MAE, and R² metrics. XGBoost’s proven track record in public-health forecasting makes it an ideal choice for life-expectancy prediction and establishes a strong baseline for future comparative work.

### 3.3.1 XGBoost

XGBoost, or eXtreme Gradient Boosting, is a powerful and efficient machine learning algorithm designed to enhance the performance of gradient boosting models. By leveraging sequential decision trees and optimizing their outputs, XGBoost iteratively refines predictions, making it an essential tool for both research and real-world applications. Its robust features, including scalability, flexibility, and efficiency, have made it a popular choice in machine learning competitions and industry use cases [7] .

As illustrated in **Fig. 4**, XGBoost represents the evolution of boosting techniques, building on the strengths of decision trees, bagging, and gradient boosting to deliver optimized gradient boosting. This evolutionary process demonstrates how XGBoost refines its approach to achieve greater efficiency and predictive power.

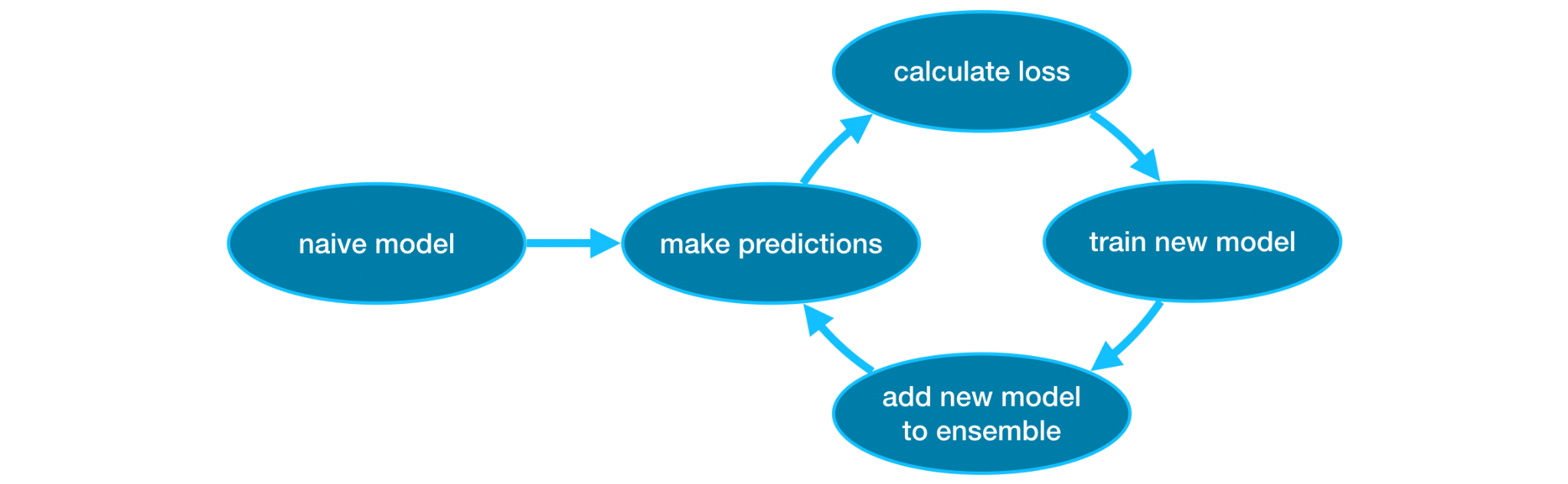
XGBoost’s primary strength lies in its ability to convert weak learners, in our case a decision tree, into a strong predictive model through an iterative boosting process. Each subsequent tree focuses on correcting the residual errors from the previous iterations, allowing the model to capture intricate patterns in the data. This step-by-step refinement ensures that the final ensemble model is both accurate and robust.

In addition to its boosting mechanism, XGBoost is known for its computational efficiency and scalability. It employs parallel processing and memory-efficient techniques, enabling it to handle large datasets with high-dimensional features effectively. Another crucial feature is its ability to automatically handle missing data, allowing users to train models without extensive preprocessing. To prevent overfitting, XGBoost support L1 (lasso) and L2 (ridge) regularization, which penalize overly complex models.

# 

**Fig. 4 - Evolution of XGBoost: From Decision Trees to Optimized Gradient Boosting [35]**

#### 3.3.1.1 Flow of the XGBoost Model



**Fig. 5 - Workflow of the XGBoost Model Showing Iterative Training and Loss Minimization [26]**

The flow of the XGBoost model involves several key steps, starting from defining the objective to iteratively building trees that enhance prediction accuracy. **Fig. 5** illustrates the workflow of the XGBoost model, highlighting its iterative process from making initial predictions to adding new models to the ensemble and minimizing loss through repeated refinements. Below is a detailed breakdown of this process:

**Define the Objective Function:** XGBoost minimizes a loss function by adding trees sequentially. For regression tasks, the loss function is typically the **Mean Squared Error (MSE)**, which quantifies the difference between the actual values and the predicted values.

The MSE is expressed as:

Here, is the Actual value and Predicted value, Land is the loss function, specifically the Mean Squared Error in this case.

This formulation ensures that XGBoost iteratively adjusts its predictions to minimize the squared errors between and, improving the accuracy of the model with each additional tree.

**Initialize the Model:** Start with an initial prediction, typically the mean value of the target variable for regression tasks: , this serves as the baseline for subsequent iterations.

**Calculate Residuals:** To improve predictions, XGBoost computes the residuals between the actual values and the current predictions. The residuals represent the direction and magnitude of adjustment needed to minimize the loss. These are calculated as the negative gradient of the loss function with respect to the predicted values:

For regression tasks, such as those using the Mean Squared Error (MSE) as the loss function, the residuals simplify to:

Here, is the loss function, specifically the Mean Squared Error, is the actual target value, is the predicted value at iteration t, and is the residual at iteration t.

The residuals serve as the basis for the next tree in the sequence, allowing the model to iteratively reduce errors and improve its predictions. By incorporating this gradient-based approach, XGBoost systematically refines the model’s accuracy.

**Fit a Tree to the Residuals:** To predict the residuals, a decision tree is constructed. This tree identifies patterns in the data that previous predictions missed, enabling the model to refine its performance. Each leaf node in the tree corresponds to a predicted residual value.

**Compute Leaf Values:** Once the tree is built, the next step is to compute the optimal values for each leaf node that minimize the loss function. This involves calculating the weighted sum of gradients and applying regularization penalties to avoid overfitting and overly complex trees.

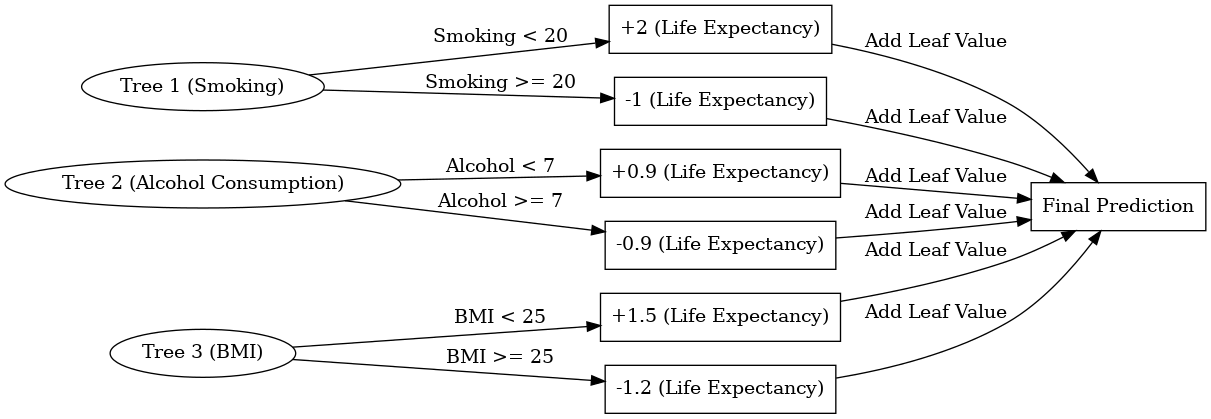
**Update Predictions:** Update the model’s predictions by adding the contribution of the new tree:

Here , is the learning rate (controls the contribution of each tree) and is the prediction of the new tree for input .

**Iterate Through Multiple Trees:** The process of calculating residuals, fitting trees, and updating predictions is repeated for a predefined number of iterations (K) or until a stopping criterion is met. This iterative approach enables the model to continuously refine its predictions.

**Generate Final Predictions:** After completing all iterations, the final prediction is obtained by summing the outputs of all trees, weighted by the learning rate:

#### 3.3.1.2 Example: Step-by-Step Flow of the XGBoost Model



**Fig. 6 - Simulation of XGBoost Using Smoking, Alcohol Consumption, and BMI to Predict Life Expectancy**

Here’s a simulation of how XGBoost works using the indicators of **Smoking**, **Alcohol Consumption**, and **BMI** to predict **Life Expectancy**. Each step includes the **formulas** for better understanding.

**Fig. 6** illustrates how XGBoost predicts life expectancy by iteratively refining predictions through a series of decision trees. The process begins with an initial prediction, such as the mean life expectancy in the training data, set at 70 years. Each tree contributes adjustments to this initial prediction based on specific decision rules derived from features like Smoking, Alcohol Consumption, and BMI.

To calculate this step-by-step, consider the following example data:

• **Smoking**: 15 (Smoking < 20, so the tree adds +2 to the prediction)

• **Alcohol Consumption**: 6 (Alcohol < 7, so the tree adds +0.9 to the prediction)

• **BMI**: 24 (BMI < 25, so the tree adds +1.5 to the prediction)

The first step is to calculate the residuals for each observation. Residuals are defined as the difference between the actual observed values and the predicted values. They play a critical role in guiding the corrections made by subsequent trees. For instance, if the actual life expectancy for a data point is 73 years and the initial prediction is 70, the residual is calculated as:

This residual informs the first tree about the error it needs to address.

**In the first iteration**, the model trains a decision tree to minimize the residuals. The first tree uses the feature Smoking and applies the decision rule: if Smoking is less than 20, the tree adds +2 to the prediction; otherwise, it subtracts 1. For the given example, Smoking is 15, so . The updated prediction after this tree is expressed as:

The new residual is then recalculated as:

**In the second iteration**, the model focuses on the residuals left by the first tree. The second tree uses the feature Alcohol Consumption and applies the decision rule: if Alcohol Consumption is less than 7, the tree adds +0.9 to the prediction; otherwise, it subtracts 0.9. For the given example, Alcohol Consumption is 6, so . The updated prediction becomes:

The residual is updated again as:

**Finally**, the third tree addresses the remaining residuals using the feature BMI. The decision rule for this tree is: if BMI is less than 25, the tree adds +1.5 to the prediction; otherwise, it subtracts 1.2. For the given example, BMI is 24, so . The updated prediction becomes:

Equivalently**,** The final prediction is achieved by summing the contributions from all trees :

This iterative process ensures that each tree incrementally reduces the error by focusing on the residuals from the previous step. By combining the initial prediction with the adjustments from all three trees, the model produces a final prediction , which is closer to the actual value of 73. This example demonstrates the effectiveness of XGBoost in refining predictions through its boosting mechanism.

**In conclusion**, by employing machine learning approaches, this research seeks to provide a comprehensive evaluation of the methodologies best suited for predicting life expectancy.

# 4. Evaluation

We begin by validating our synthetic life-expectancy labels against known population benchmarks. First, we examine the overall distribution of generated lifespans: the mean should lie close to 78.4 years [33], with the bulk of values falling between 70 and 80. Next, we compute indicator-specific means to verify realistic relationships—for example, the average life expectancy of smokers must be lower than that of non-smokers, and those reporting regular physical activity should exhibit higher mean lifespans than sedentary individuals. We apply the same comparative checks across all risk and protective factors to ensure that our synthetic data faithfully reflect established epidemiological patterns.

The XGBoost model is evaluated using standard regression metrics—Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R²) (see Section 4.1)—which together quantify prediction bias, error magnitude, and explained variance. Below, we outline our evaluation protocol and describe each metric in detail.

## 4.1. Metrics in Detail

**Mean Absolute Error (MAE) -** MAE measures the average magnitude of errors between the predicted and actual values, regardless of their direction. It is a straightforward metric that provides a clear understanding of how far predictions deviate from the true values on average. MAE treats all errors equally, making it robust against the influence of large outliers. This simplicity makes it a reliable choice for comparing models, especially when a balanced view of prediction accuracy is desired. The formula used **,** where is the actual value, the predicted value, and the number of observations. The absolute value ensures that the errors are not canceled out by their direction.

**Root Mean Squared Error (RMSE) -** RMSE measures the square root of the average squared differences between the predicted and actual values. Unlike MAE, RMSE gives more weight to larger errors, penalizing them more heavily. This makes RMSE particularly useful when large prediction errors are more concerning than small ones. RMSE provides an intuitive sense of error magnitude, as it is expressed in the same unit as the target variable, making it easy to interpret. The formula used is  , where and represents the actual and predicted values, respectively, and is the total number of observations. The squaring amplifies larger errors, while the square root brings the result back to the same scale as the original data.

**R-squared (R²) -** R² quantifies the proportion of variance in the dependent variable that is explained by the independent variables in the model. It provides a measure of how well the model fits the data, with values ranging from 0 to 1. An R² value close to 1 indicates that the model explains most of the variability in the target variable, while a value near 0 suggests limited explanatory power. R² is particularly useful for comparing models to see which one better captures the underlying patterns in the data. The formula used is , where represents the actual values, the predicted values, and the mean of the actual values. The numerator calculates the residual sum of squares (unexplained variance), while the denominator calculates the total sum of squares (total variance). The resulting value represents the proportion of variance explained by the model, with higher values indicating better performance.

These metrics together provide a comprehensive evaluation framework. While MAE offers a straightforward view of average errors, RMSE highlights significant deviations, and R² indicates the overall explanatory power of the model. XGBoost is rigorously evaluated using these metrics, providing comprehensive insights into their predictive capabilities. XGBoost excels in efficiency and feature importance analysis, The use of these metrics ensures that the models’ predictions are not only accurate but also interpretable and actionable.

# 5. Project Architecture and Deployment

# 

**Fig. 7 - Project Architecture and Deployment Workflow**

This project is designed to predict life expectancy based on user-provided information through an interactive web application. Below is a detailed overview of the system architecture, components, and their roles.

**Frontend:**

**Next.js [27]** with **React [29]** and **TypeScript [30]** are used to create an interactive and maintainable web application with fast rendering and dynamic data handling.

**Tailwind CSS [28]**: Simplifies styling with utility-first classes, enabling quick development of responsive and visually appealing designs.

**Backend**

**Flask**: A lightweight and flexible Python web framework for building web applications and APIs. It provides essential tools and libraries while allowing developers to customize components, making it ideal for simple to moderately complex projects [32].

**XGBoost**: A library for gradient boosting, included to handle tabular data problems where tree-based models perform better than deep learning [22].

**Pandas**: Handles data preprocessing and manipulation and preparing user inputs for ML models.

To ensure seamless interaction between the front end and the machine learning backend, the application is deployed across two different platforms:

**Frontend Deployment – Hosted on Vercel**: Vercel is optimized for hosting static and server-rendered front-end applications, enabling fast global delivery, continuous integration, and automatic deployment with Git support.

**Backend Deployment – Hosted on Amazon EC2**: Amazon EC2 (Elastic Compute Cloud) provides scalable compute capacity and gives developers full control over the hosting environment. The Flask API is exposed over the internet to handle POST requests from the front end and return predictions.

**Why This Architecture?**

Next.js and React provide a robust structure for building dynamic, interactive UIs, while TypeScript ensures code maintainability by reducing runtime errors. Tailwind CSS accelerates styling workflows.

Flask offers simplicity, flexibility, and ease of integration for deploying machine learning models. TensorFlow and PyTorch provide flexibility and reliability for ML tasks, and Pandas efficiently preprocesses data.

This architecture ensures fast response times, robust data handling, and smooth integration between the frontend and backend.

**Overall Process**

This project predicts life expectancy based on user input through the following steps:

1. User Input: Users submit data via a form on the frontend.
2. Data Transmission: The frontend sends validated data to the backend API.
3. ML Prediction: The backend processes the data, applies preprocessing, and uses machine learning models to predict life expectancy.
4. Response: The backend sends the prediction back to the frontend, where it is displayed to the user interactively and intuitively.

# 6. Results

This section presents the final outcome of our modeling process: the development and evaluation of an XGBoost model for predicting life expectancy using our realistic synthetic dataset. In the following results, "actual" values refer to the ground truth labels from our generated synthetic data, while "predicted" values are the model's forecasts based on the input features.

We trained our XGBoost regressor using a grid-search over key hyperparameters and selected the following best configuration to minimize validation error:

Best Parameters: {

colsample\_bytree: 0.5442,

gamma: 0.09799,

learning\_rate: 0.0235,

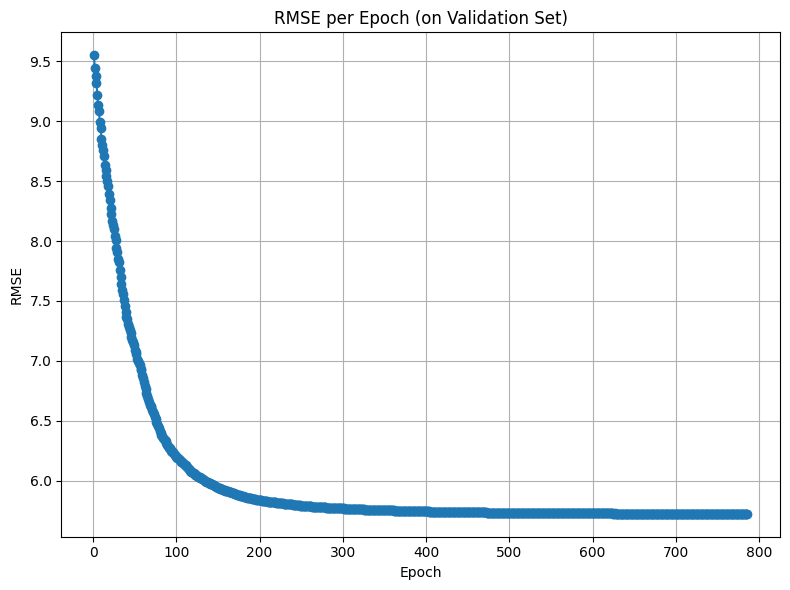
max\_depth: 7,

n\_estimators: 819,

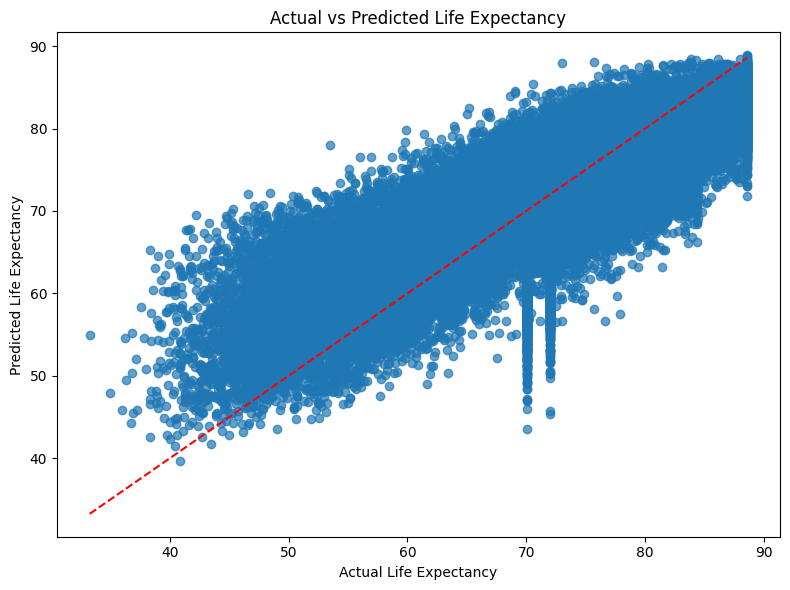
subsample: 0.8736 }

Under our optimized XGBoost configuration, the model achieves an RMSE of 5.73 years, meaning that its predictions deviate from the true life-expectancy values by an average of roughly ±5.7 years. The R² score of 0.65 indicates that about 65% of the variability in life expectancy across individuals is explained by our selected personal, lifestyle, and medical features. On average, the absolute error (MAE) is 4.4 years, so half the time our forecast is within 4½ years of the actual value. The mean absolute percentage error (MAPE) of 6.6% shows that relative to each person’s true longevity, our predictions are off by under 7% on average. Finally, the explained variance score (0.65) echoes the R² result, confirming that two-thirds of the signal in the data is captured by the model.

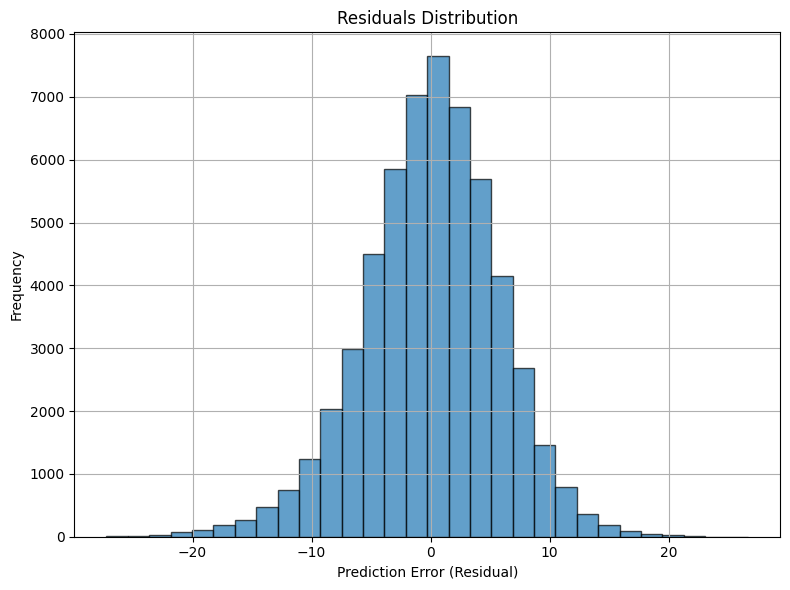
As shown in **Fig. 8**, the RMSE steadily decreases over 700 epochs, converging around 5.73 by epoch 796. **Fig. 9** plots actual versus predicted life expectancy on the hold-out set, illustrating a strong positive correlation with some scatter. The residuals (prediction errors) are approximately normally distributed around zero—see **Fig. 10**—indicating no major bias. Finally, **Fig. 11** examines residuals against predicted values, confirming homoscedasticity across the prediction range.



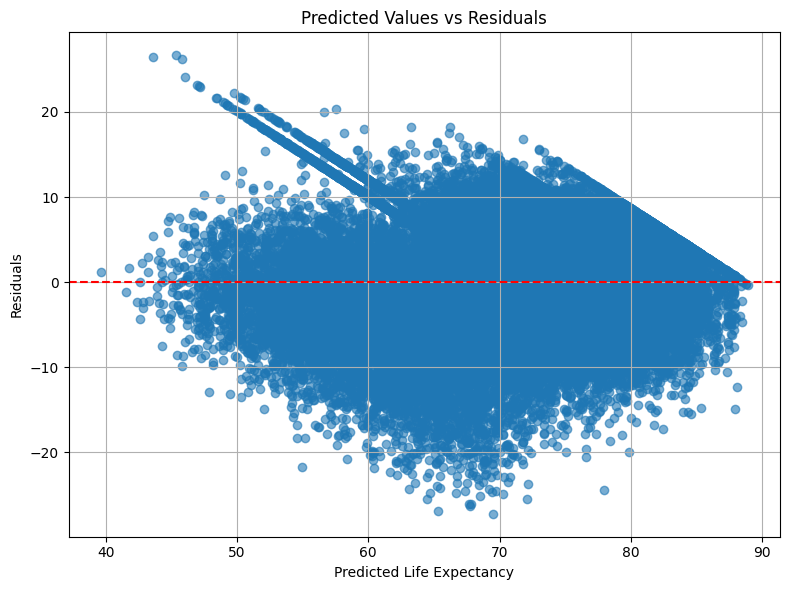
**Fig. 8 - RMSE per Epoch on Validation Set**



**Fig. 9 - Actual vs. Predicted Life Expectancy**



**Fig. 10 - Residuals Distribution**



**Fig. 11 - Predicted Values vs. Residuals**

# 7. User Guide

This section provides a comprehensive overview of how users can interact with the Life Expectancy Calculator web application. It is designed to ensure that users of varying backgrounds and technical skills can easily navigate and benefit from the system. The guide outlines the platform’s key features, including multilingual support, personalized health tips, and an interactive life expectancy simulator. In addition, visual walkthroughs with annotated screenshots are included to explain each part of the website and demonstrate how users can input data, interpret their results, and explore different health scenarios. Together, these tools aim to enhance accessibility, encourage informed decision-making, and promote health awareness through an intuitive and user-friendly experience.

## 7.1 Key Features That Enhance User Experience

This project includes several important features designed to improve usability and user engagement:

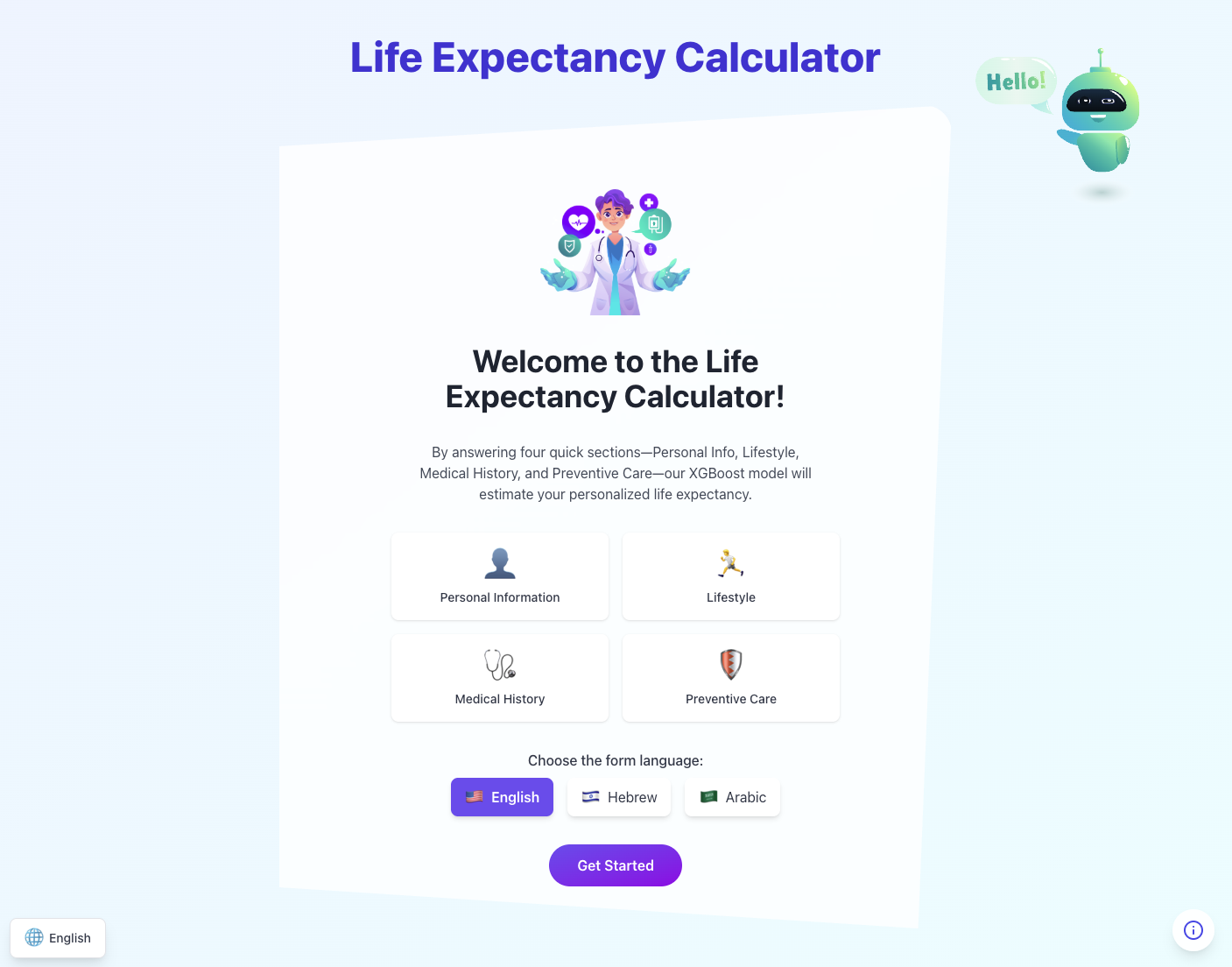
* **Multilingual Support:** The website supports three languages—Arabic, Hebrew, and English. This is especially important given that some medical terms and illness names are commonly known only in English and can be difficult for users to understand. Supporting multiple languages ensures the platform is accessible and useful to a broader audience.
* **Personalized Health Tips**: Alongside the estimated life expectancy, the website provides personalized health tips based on the user’s input data. These tips highlight how specific negative health indicators may be impacting the user’s life expectancy and suggest areas for improvement.
* **Interactive Life Expectancy Simulator [Fig. 19]** : One of the most impactful features is the simulator. It allows users to modify changeable health-related data—such as losing weight, quitting smoking, or starting physical activity—to visualize how these changes could positively affect their life expectancy. This empowers users to make informed lifestyle decisions.

## 7.2. Explanation with screenshots

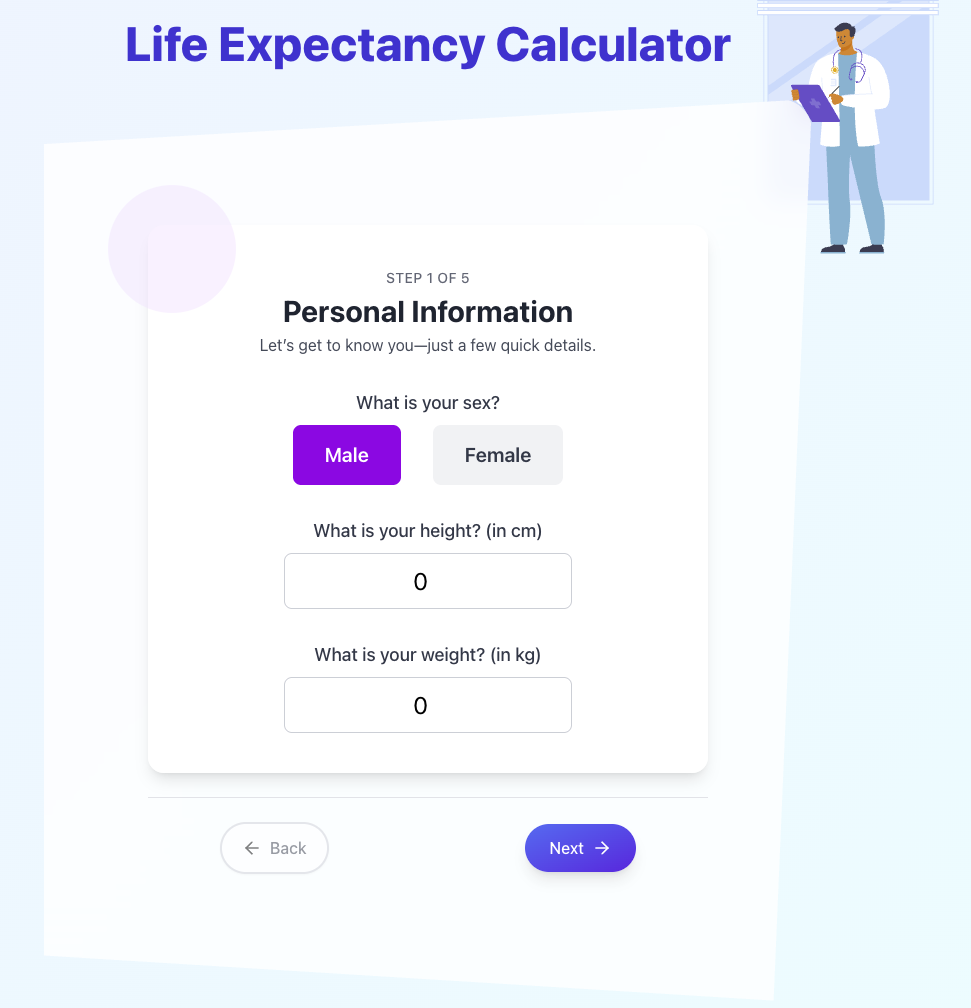
The system predicts individual life expectancy based on health-related indicators, using an XGBoost model trained on synthetic data.

**Explanation with Screenshots :**

The landing screen (**Fig. 12**) presents a clean and user-friendly interface that welcomes users to the Life Expectancy Calculator. It features a bold and clear title, a friendly health-themed illustration that adds a professional and approachable feel, and a concise introduction explaining the four main sections: Personal Information, Lifestyle, Medical History, and Preventive Care. Each section is represented by an icon-enhanced tile to improve navigation and clarity. A prominent “Get Started” button invites users to begin the process immediately. The interface also includes multilingual support, allowing users to choose their preferred language—English, Hebrew, or Arabic—through a language selector displayed above the “Get Started” button. Additionally, a persistent language button is located in the bottom-left corner of the screen, enabling users to switch languages at any point during their interaction. These design choices enhance accessibility and ensure that the platform is easy to use for a diverse audience.

****

**Fig. 12 – Home Page of the Life Expectancy Calculator**

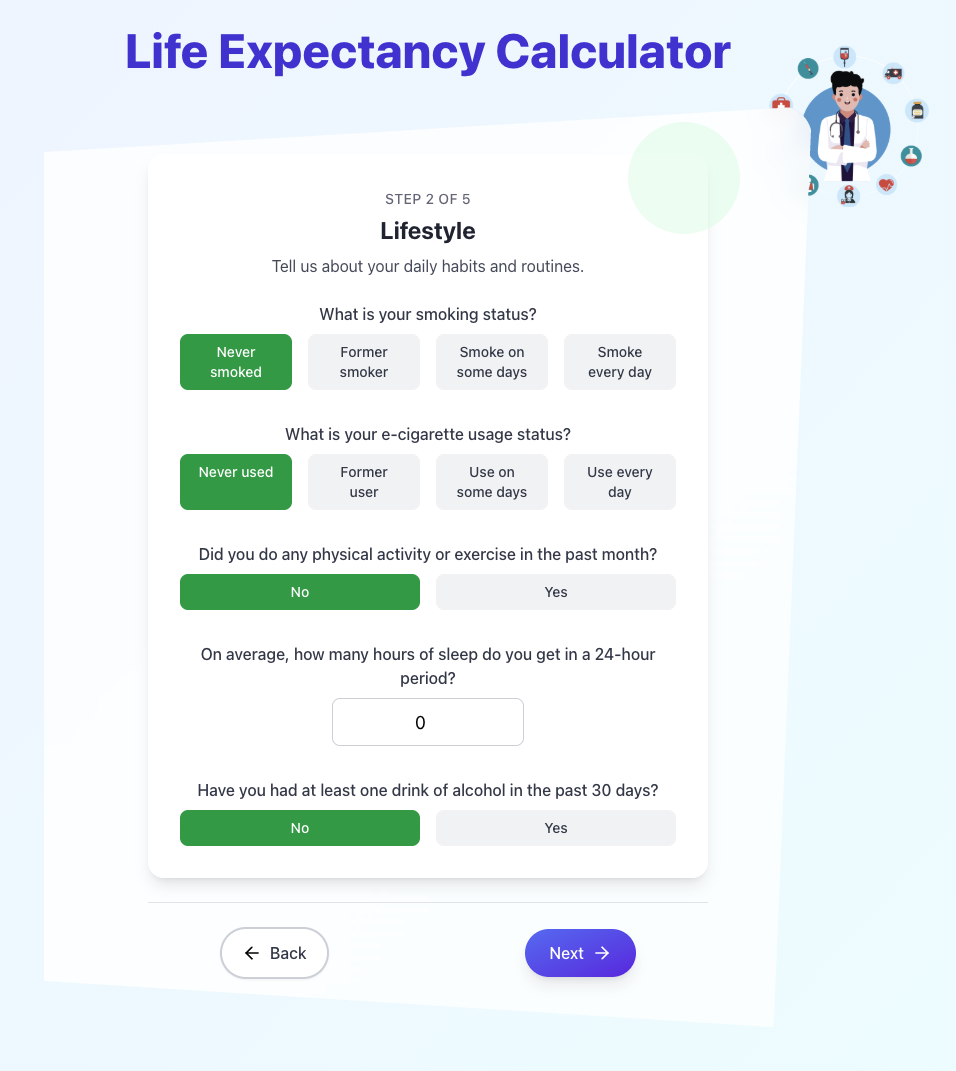
****

**Fig. 13.1 – Personal Information (Step 1 of 5)**

Next, we have the form, which consists of five sections, starting with Personal Information (**Fig. 13.1**). On this screen, you’ll provide three basic data points that help tailor your life expectancy estimate:

1. Sex : Tap Male or Female to indicate your sex.
2. Height : In the “What is your height?” field, type your height in centimeters (e.g. “170”).
3. Weight : In the “What is your weight?” field, type your weight in kilograms (e.g. “65”).

Once all three fields are filled, press the Next button to move on to Step 2.

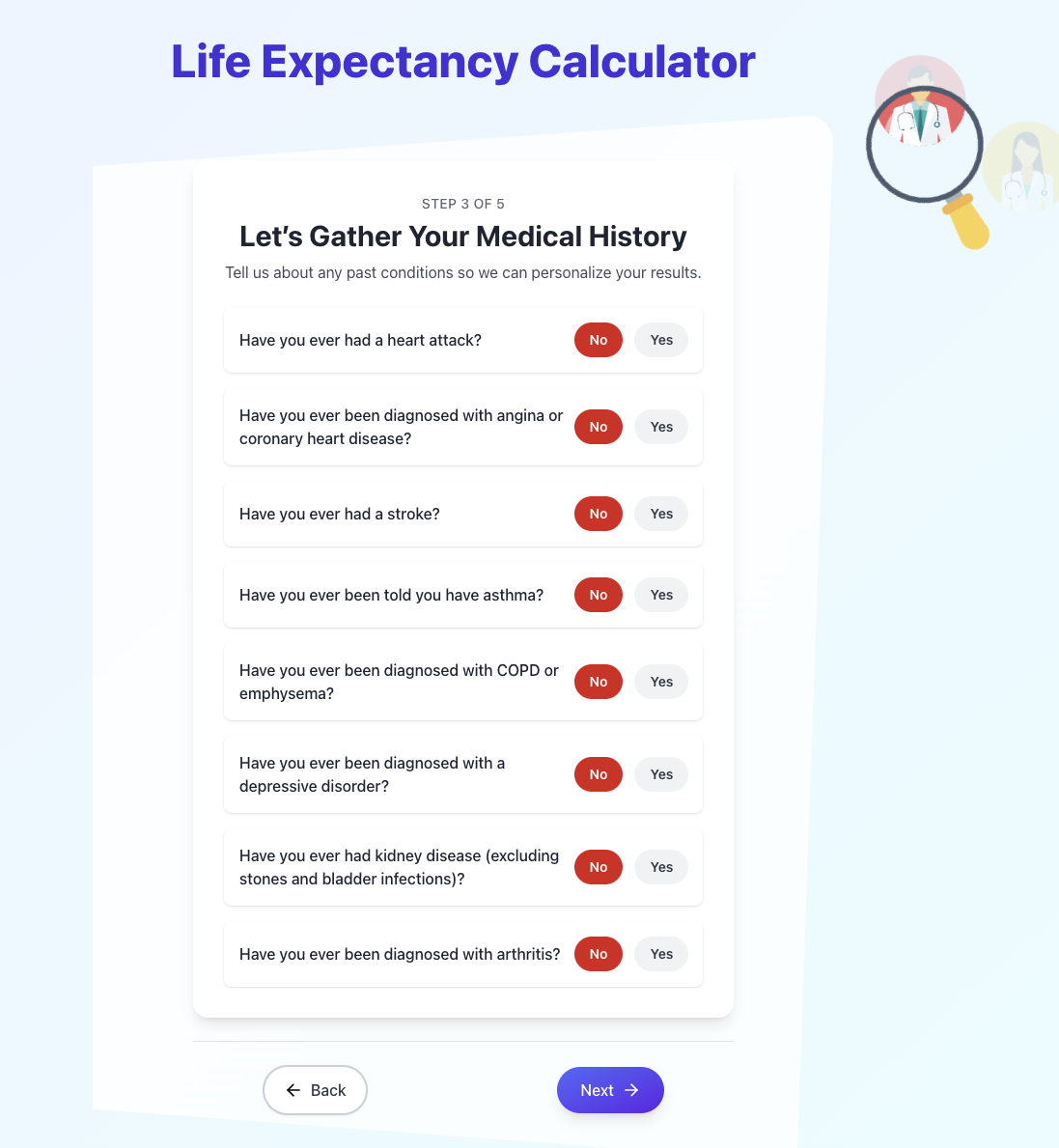
****

**Fig. 13.2 – Lifestyle (Step 2 of 5)**

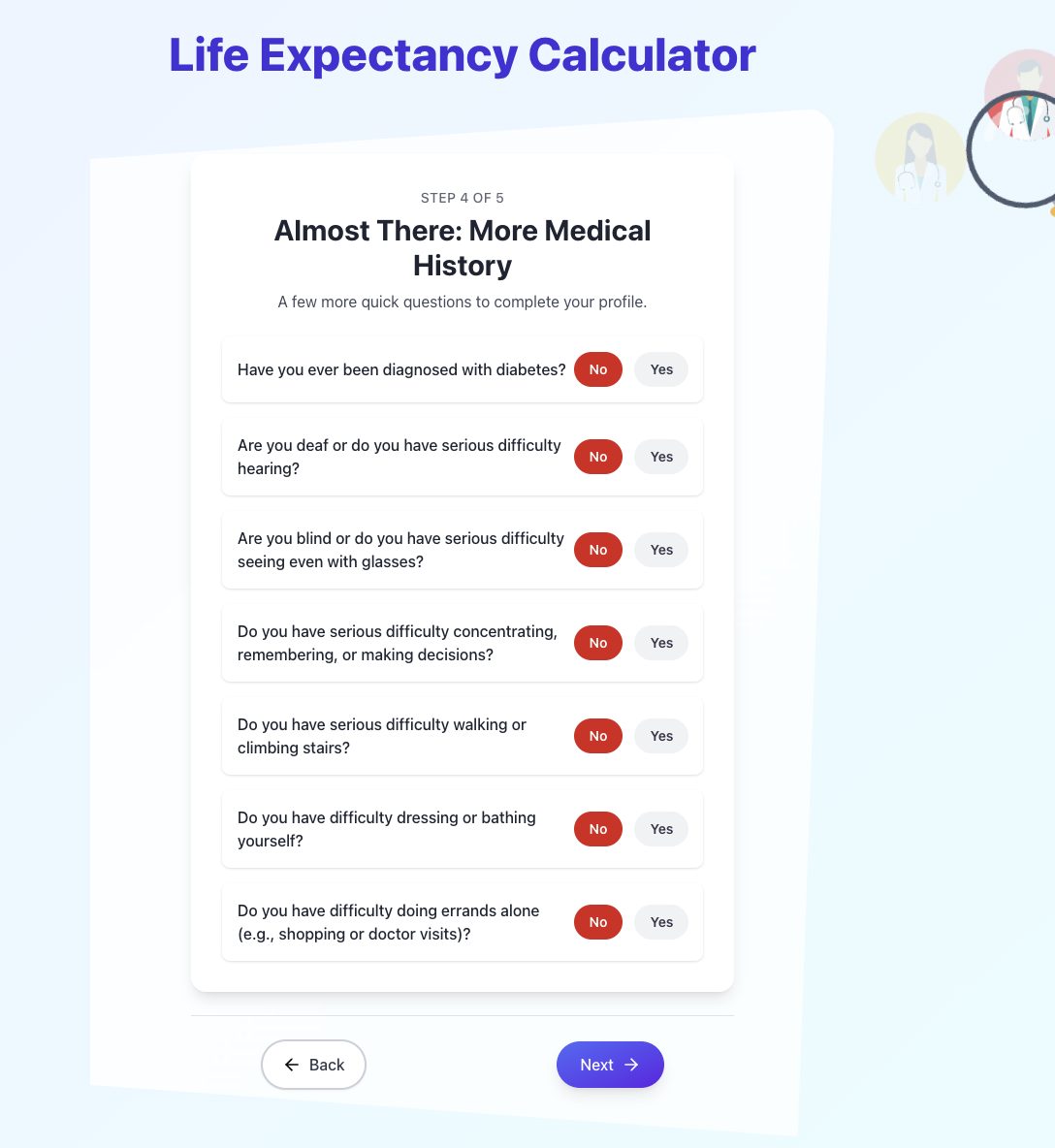
On this screen **Fig. 13.2** you’ll report daily habits that affect your health:

1. **Smoking Status** : Tap one of the four options to describe your cigarette use  
    - Never smoked  
    - Former smoker  
    - Smoke on some days  
    - Smoke every day
2. **E-Cigarette Usage** :Select one of four to indicate your vaping history:  
    - Never used  
    - Former user  
    - Use on some days  
    - Use every day
3. **Physical Activity** : Choose Yes or No to tell us if you’ve exercised or done any physical activity in the past month.
4. **Sleep Hours** : In the numeric field, enter the average number of hours you sleep within a 24-hour period (for example, “7”).
5. **Alcohol Consumption** : Select Yes or No to indicate whether you’ve had at least one alcoholic drink in the past 30 days.

Once all answers are selected or entered, press Next to continue to the Medical History section—or tap Back to change your Personal Information.

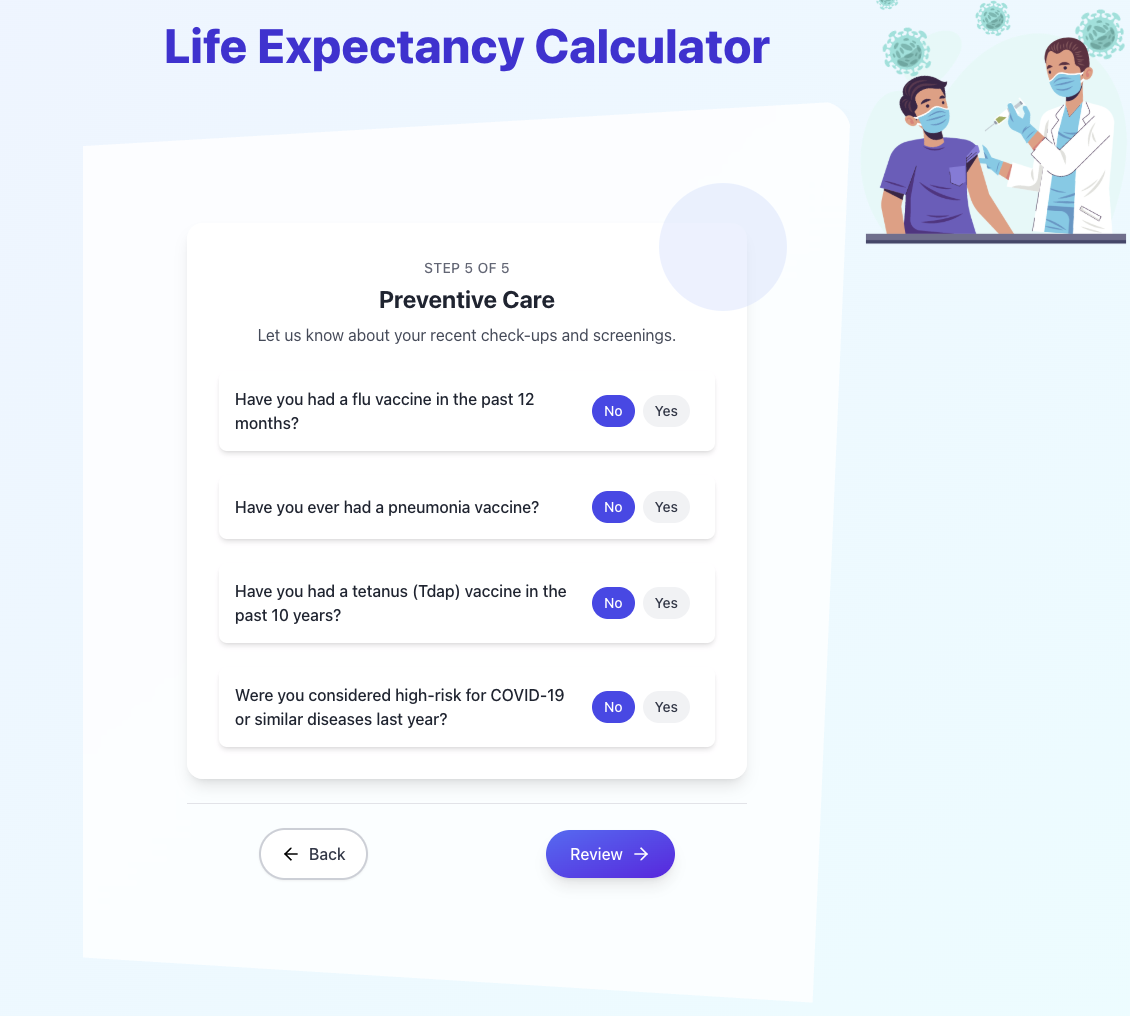
****

**Fig. 13.3 - a : Medical History (Step 3 of 5)**



**Fig. 13.3 - b : Medical History (Step 4 of 5)**

On this screen **Fig.13.3** a and b you’ll review a series of 15 yes/no questions about past diagnoses and health conditions (e.g. heart attack, stroke, asthma, diabetes, and more). Simply scroll through the list, tap Yes or No for each item, and then press Next when you’ve answered them all. If you need to change your previous responses, use the Back button.

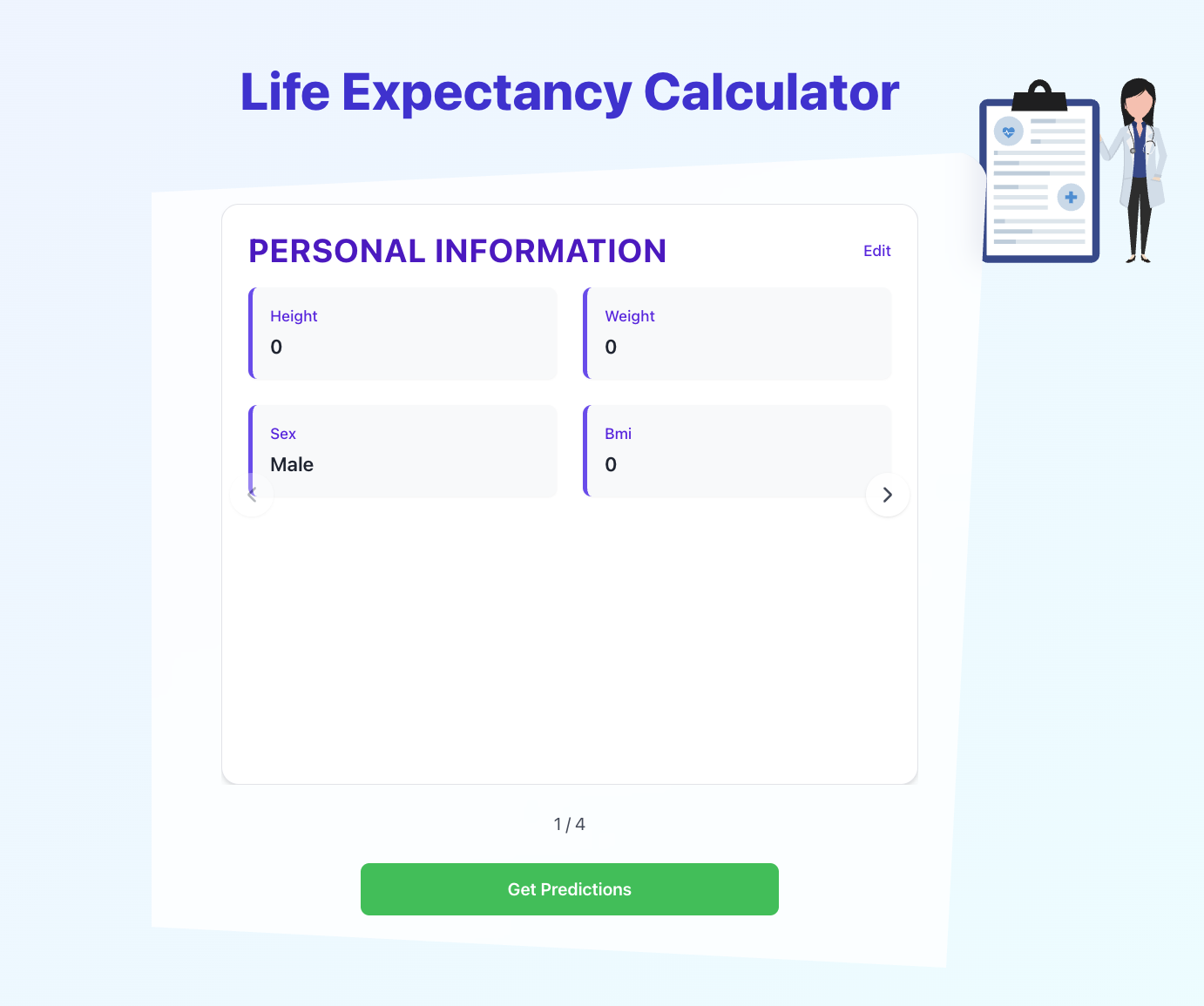
****

**Fig. 13.4 – Medical History (Step 5 of 5)**

On this final section **Fig.13.4** you’ll answer four yes/no questions about recent vaccinations and risk status:

1. Flu Vaccine : Select Yes if you received a flu shot within the past 12 months.
2. Pneumonia Vaccine : Select Yes if you’ve ever had a pneumonia vaccine.
3. Tetanus (Tdap) Vaccine :Select Yes if you’ve had a tetanus booster in the last 10 years.
4. High-Risk Status :Select Yes if you were considered high-risk for COVID-19 or similar diseases during the last year.

When all four questions are answered, tap Review to see your summary before submitting. Use Back if you need to change any previous answers.

****

**Fig. 14 - Review Form Details**

After completing all four steps, you’ll land on the Review screen **Fig.14** where you can verify every answer before getting your life-expectancy prediction:

1. Section Carousel  
   * At the top of the card you’ll see the current section title (e.g. “PERSONAL INFORMATION”).
   * A page indicator below shows “1 / 4” (or whichever section you’re viewing).
2. Navigate Between Sections  
   * Use the left and right arrow buttons at the sides of the card to scroll through each of the four summary panels.
3. Edit Any Section  
   * Click the Edit link in the top-right of the card to jump back directly to that section’s form and adjust your answers.
4. Submit for Prediction  
   * Once you’ve confirmed all data, press the large green Get Predictions button.
   * Your responses will be sent to our model, and you’ll immediately see your personalized life-expectancy estimate.

This review step ensures you have one last chance to correct typos or mis-clicks before generating your final result.

****

**Fig. 15 – Result : Predicted Life Expectancy**

After you tap Get Predictions, our model may take 10–15 seconds to process your inputs. A loading indicator will appear during this time; please wait until the results card displays **Fig.15** .

Your Estimated Life Expectancy

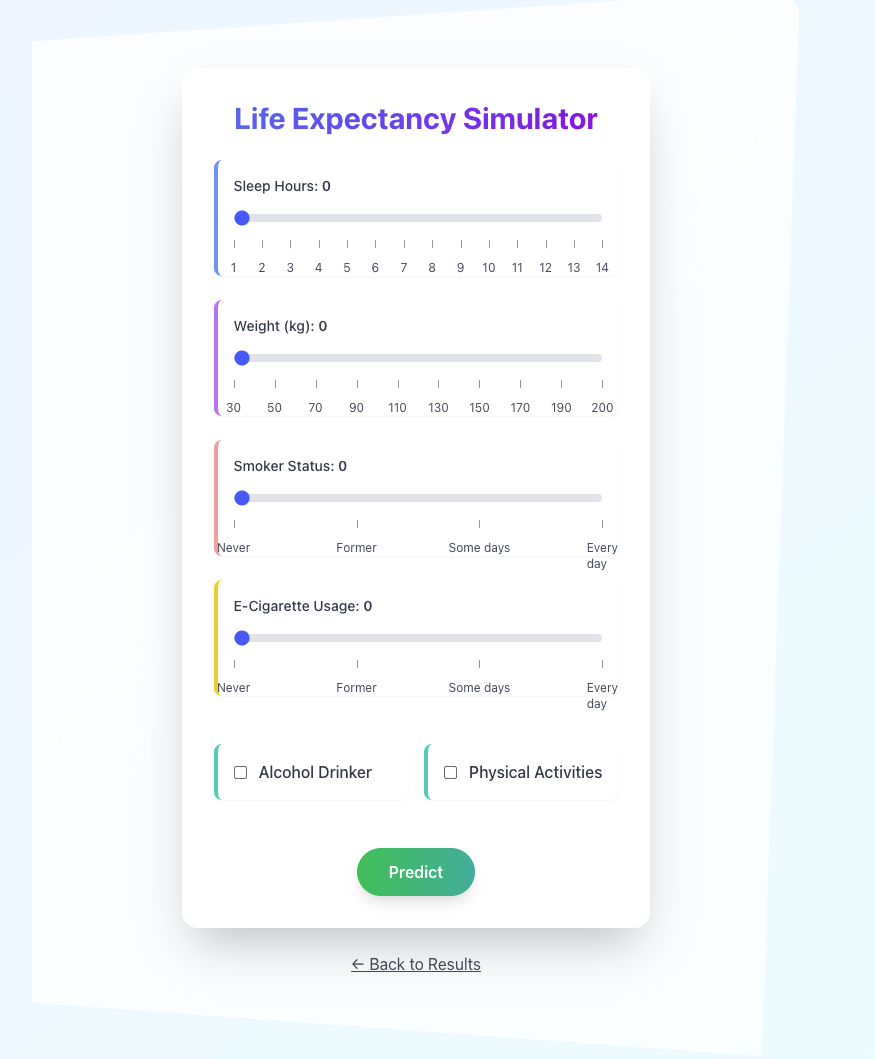
* At the center of the card you’ll see your personalized life-expectancy estimate in large green type (e.g. 62.8 years).

Personalized Health Tips

* Below the estimate is a set of concise, color-coded tips showing which factors—such as BMI, sleep, exercise, and vaccinations—are associated with gains or losses in life expectancy.

Next Actions

* Try the Simulator(**Fig. 16**): adjust any answer directly on this page to see how changes could increase your predicted lifespan.
* Start Over: clear all entries and return to Step 1 to run a completely new calculation.

****

**Fig. 16 – Life Expectancy Simulator**

After viewing your initial result, you can experiment with different values to see how changes in your habits and measurements affect your estimated lifespan:

1. Sleep Hours :Drag the slider to set your average nightly sleep (1–14 hours).
2. Weight (kg) : Use the slider to adjust your weight between 30 kg and 200 kg.
3. Smoker Status : Slide between four positions to simulate cigarette use.
4. E-Cigarette Usage : Slide between four positions to simulate e-cigarette use.
5. Alcohol Drinker : Check this box if you drink alcohol; uncheck to simulate quitting.
6. Physical Activities : Check this box if you exercise regularly; uncheck to simulate a more sedentary lifestyle.

When you’ve adjusted the sliders and checkboxes, tap Predict to recalculate your life-expectancy estimate instantly. This interactive tool helps you explore how small changes—like getting more sleep or quitting smoking—can boost your predicted lifespan.

# 8. Maintenance Guide

This section serves as a comprehensive technical reference for developers who wish to install, modify, or extend the Life Expectancy Calculator system. It outlines the full development environment, installation steps, and the architecture behind both backend and frontend components. Additionally, it provides guidance on synthetic data generation, model training, and translation support. Whether deploying the project for the first time or building upon its core features, this guide ensures that future developers can efficiently maintain and scale the system.

## 8.1 Environment

* **Backend:**
  + Python environment with dependencies: Flask, XGBoost, Pandas, NumPy, Scikit-learn.
  + AWS EC2 for deployment.
* **Frontend:**
  + Next.js framework with React, TypeScript, and Tailwind CSS.

## 8.2 Installation Instructions:

1. **Backend Setup:**
   * Clone the backend repository: (<https://github.com/mahersalman/life-expectancy/tree/main/ML-Server>)
   * Install dependencies with pip install -r requirements.txt.
   * Deploy Flask application locally (python app.py) or via AWS EC2 (ensure proper AWS setup and configurations).
   * Ensure .pkl files (model and scaler) are correctly placed in the Flask application's directory.
2. **Frontend Setup:**
   * Clone the frontend repository: <https://github.com/mahersalman/life-expectancy/tree/main/life-expectancy-app>
   * Navigate to the project directory and run npm install.
   * Launch application with npm run dev for local development or build for production deployment with npm run build.

**Note:**

* If your API is hosted on AWS EC2 (or any other remote server), update the destination URL in your vercel.json file to point at that host.
* If you’re running the server locally, set SERVER\_URL to http://localhost:3000 (or whatever port you’ve configured).

## 8.3 Synthetic Data Generation & XGBoost Training

This part of the project is organized into two core modules—data generation and model training—each encapsulated in its own notebook for clarity and reproducibility:

1. **Synthetic Data Generation** The GenerateData.ipynb notebook (located in Data/) drives the creation of our individual‐level dataset. It ingests:  
   * **heart\_2022\_with\_nans.csv**—the raw, person-level records containing missing entries
   * **xmart.csv**—country-year life-expectancy averages
2. Through a pipeline of imputation, encoding, normalization, and probabilistic sampling, this script rebuilds a fully synthetic cohort that mirrors real-world distributions. You can easily toggle which indicators to include or exclude—just update the indicator list at the top of the notebook and rerun.
3. **XGBoost Model Training** The Xgboost.ipynb notebook (under ML-Server/) handles everything from loading the synthetic data to exporting your final model:  
   * **Data Loading & Path Configuration** — Point the notebook at your local or cloud drive path.
   * **Hyperparameter Tuning** — Includes grid-search routines for tree count, depth, learning rate, and subsampling.
   * **Training & Evaluation** — Trains on the 70% split, validates on 15%, and tests on the remaining 15%, reporting RMSE, MAE, and R².
   * **Model Export** — Once satisfied with performance, save the fitted XGBoost object as a .pkl file and commit it to your repository for seamless integration with the Flask API.

Together, these notebooks form a reproducible pipeline: regenerate data as your indicators evolve, retrain XGBoost in minutes, and deploy directly to your backend.

## 8.4 Life Expectancy Next.js Website (Front-End)

This Next.js app provides the user interface for collecting inputs and displaying life-expectancy predictions. To get started locally:

npm install

npm run dev

Then point your browser to http://localhost:3000.

### 8.4.1 Entry Point (src/app/page.tsx)

This client-side component sets up the app’s full-screen gradient layout, animated header, and Lottie mascot, then renders all routes inside a clipped, semi-transparent card. It uses React Router to switch between:

* **/** → HomePage
* **/form** → UserForm
* **/review** → Review
* **/result** → Results
* **/simulator** → Simulator

### 8.4.2 Core Components

* **src/components/HomePage.tsx:** A welcome/landing screen that introduces users to the tool.
* **src/components/UserForm.tsx :** Orchestrates the multi-step questionnaire, advances through each section, and submits the final payload.
* **Section-Specific Forms:**
  + **PersonalInfoForm.tsx** — captures basic demographics (sex) and anthropometrics (height & weight).
  + **LifestyleForm.tsx** — gathers habits and exposures (smoking & e-cigarette use, exercise, sleep hours, alcohol intake).
  + **MedicalHistoryFormPart1.tsx** — records major cardiovascular & respiratory conditions (heart attack, angina/CHD, stroke, asthma, COPD) and mood/kidney/arthritis history.
  + **MedicalHistoryFormPart2.tsx** — continues with metabolic, sensory, cognitive & mobility issues (diabetes, hearing/vision difficulty, concentration challenges, walking/climbing, self-care & errands).
  + **PreventiveCareForm.tsx** — asks about vaccinations, cancer screenings, and other preventive health checks.
* **src/components/Review.tsx** & **ReviewCard.tsx:** Display a summary of entered answers and allow users to jump back and correct any step.
* **src/components/Results.tsx & DisplayTips.tsx :**Renders the life-expectancy estimate fetched from the backend alongside personalized health tips. After displaying the predicted LE value, it imports and uses the Tips.ts module to surface actionable advice tailored to the user’s input—highlighting factors that most influence their forecasted lifespan and suggesting ways to improve it.
* **Language Selector (src/components/LanguageSelector.tsx):** Renders a language selection interface using emoji-labeled buttons, allowing users to switch between English, Hebrew, and Arabic. Utilizes Framer Motion for interactive button animations and accesses the current language state via useLanguage() to ensure consistent UI updates and direction handling.
* **Floating Language Selector (src/components/LanguageSelector.tsx):** Displays a fixed-position language selector in the bottom-left corner of the screen, always visible throughout the app. Includes a toggleable dropdown that allows users to change the interface language at any time. Handles outside clicks to automatically close the menu, improving usability on all screen sizes.
* **src/components/Simulator:** a mini “what-if” panel with sliders (sleep, weight/BMI, smoking, e-cigs) and toggles (alcohol, exercise) that calls fetchResult(...) to recalc and display the new estimate inline.
* **Interactive Helpers: SliderWithLabelsV2.tsx**, **ToggleWithLabel.tsx**— custom inputs and mini-simulations.

### 8.4.3 Context Providers

* **Form Context (src/context/FormContext.tsx):** Defines a React context and provider for formData and its updater, initialized from initialFormData. Components can call useFormContext() to read the current form state and call setFormData() to update it, ensuring user inputs stay synchronized across the entire app.
* **src/context/LottieContext.tsx :**Manages loading and playing of Lottie animations stored under src/Lottie/\*.json.
* **Language Context (src/context/LanguageContext.tsx)**: Provides a React context and provider for managing the selected language across the app. Initialized with English by default, it supports Arabic, Hebrew, and English, each with appropriate directionality (ltr or rtl). Components can call useLanguage() to access or update the current language, ensuring consistent localization and layout direction throughout the application.

### 8.4.4 Utility Modules

* **Question Schema (src/utils/**[**Questions.ts**](http://questions.ts)**) :**Defines a Question type and four lists—personalInfoQuestions, lifestyleQuestions, medicalHistoryQuestions, preventiveCareQuestions—each entry specifying its name, label, question, type (number with optional min/max or radio with options).
* **Health Tips (src/utils/**[**Tips.ts**](http://tips.ts)**):** Exports a tips map where each form field key links to an array of conditions and messages. At results time, the app checks each condition against the user’s data and displays matching messages—e.g., low BMI warns of reduced LE, missing flu vaccine offers a +10-year benefit, daily smoking flags an ~8.8-year reduction, etc. This drives the personalized advice panel shown alongside predictions.
* **src/utils/**[**initialData.ts**](http://initialdata.ts)**:**  Supplies default values for all form fields on first render.
* **API Utility (src/utils/**[**fetchResult.ts**](http://fetchresult.ts)**):** Defines fetchResult(formData: FormData): Promise<number>, which sends a POST to /api/predict with the user’s inputs, handles non-OK responses by extracting error details, and returns the numeric prediction field from the JSON response.

Here’s a new section **9.4.5 Translation and Language Support** written in the same style and level of detail as the previous entries, including a clear explanation of how translations are structured and how to extend support for new languages:

## 8.4.5 Translation and Language Support

* **/Translations:** This directory contains structured translation files that provide localized text for each section and component of the application. Each file exports an object keyed by LanguageType (en, he, ar), ensuring all text elements are translated and direction-aware based on the selected language. For example, resultsText.ts provides localized strings such as titles, loading messages, and button labels shown on the results page.
* **Extending Language Support**: The application currently supports English, Hebrew, and Arabic. Adding a new language requires the following four steps:  
  1. **Update the LanguageType enum** in @/app/type to include the new language code (e.g., fr for French).
  2. **Add a new configuration** in LanguageConfigs within LanguageContext.tsx, specifying the new language code and its text direction (ltr or rtl).
  3. **Include the new language** in all translation files under src/Translations/, adding the corresponding localized text values.
  4. **Extend translations for dynamic content** by updating relevant utility files in ./src/utils—such as question generators or health tip logic—to support the new language, ensuring that all text shown to users (e.g., form questions, suggestions, or feedback tips) is localized.

This modular approach ensures that adding new languages remains simple, consistent, and scalable across both static UI and dynamic content.

# 9. Usability Evaluation and Feedback Insights

To assess the usability and overall user satisfaction of the Life Expectancy Calculator, a comprehensive System Usability Scale (SUS) survey was conducted with 32 participants. The evaluation utilized a standardized questionnaire consisting of ten Likert-scale items, each designed to measure different aspects of system usability including ease of use, learnability, efficiency, and user satisfaction. The SUS scores collected ranged from 50 to 100, with an average score of approximately 78, placing the system in the "Good to Excellent" usability range according to standard SUS interpretation guidelines.

**Analysis of Individual SUS Questions Performance:**

The detailed analysis of the 32 responses revealed strong performance across most usability dimensions. Questions that received the highest scores included "I thought the system was easy to use" and "I would imagine that most people would learn to use this system very quickly," indicating excellent learnability and ease of use. Users also showed high confidence using the system and found the functions well integrated.

The system performed exceptionally well on reverse-scored questions, meaning users disagreed with negative statements about complexity, inconsistency, cumbersome use, and learning requirements. Users also indicated minimal need for technical support.

The question with the most room for improvement was "I think that I would like to use this system frequently," suggesting that while users found the system usable, they viewed it more as a one-time tool rather than something for regular use.

**Areas of Strong Performance:**

* Ease of use and intuitive design
* Quick learnability
* User confidence during interaction
* System integration and consistency
* Minimal need for technical support

**Areas Identified for Improvement:**

* Language barriers and medical terminology clarity
* Scrolling visibility issues in medical sections
* Request for personalized tips on how to increase life expectancy and understanding what factors most significantly affect individual life expectancy

The quantitative results were complemented by qualitative feedback through open-ended questions about user experience, interface clarity, perceived accuracy, and suggestions for improvement. This mixed-methods approach provided a comprehensive understanding of both what users could accomplish with the system and how they felt about the experience.

Despite the overall positive feedback, the qualitative responses revealed specific areas requiring attention. The most frequently reported issue was language-related difficulty, with multiple participants explicitly mentioning problems with "complex english in some questions". Medical terminology posed particular challenges, with users stating "the diseases are not clear to me" and "I do not know part of the illnesses and vaccines," as we can see in **Fig. 13.3** showing some difficult words related to medical terms. Additionally, users experienced interface navigation issues, particularly noting that "in medical information sections I didn't notice that there are more questions and I must scroll to see all questions," indicating visibility problems with the form design.

In direct response to this feedback, we implemented comprehensive multilingual support, adding Hebrew and Arabic alongside English. Additionally, we redesigned the medical information section shown in **Fig. 13.3a** and **Fig. 13.3b**, splitting what was previously one long scrollable section into two separate sections to improve visibility and ensure users don't miss any questions. This structural change addresses the specific navigation concern raised by users about hidden questions in scrolling sections.

A key enhancement request was to "add a section to the prediction page which includes advice for lifestyle improvements to lead to better results." In response to this feedback, we implemented a comprehensive notes section that provides personalized tips and recommendations for improving health outcomes based on the user's individual information and risk factors.

The system's responsiveness received unanimous praise, with all users describing the speed as "very fast and good," indicating strong technical performance that met user expectations for real-time interaction.

Overall, the usability evaluation confirmed the system's fundamental strengths in design and functionality while providing actionable insights that led to meaningful improvements. The integration of multilingual support based directly on user feedback demonstrates a strong commitment to accessibility and user-centered design principles. The evaluation process not only validated the system's usability but also established a foundation for continuous improvement based on real user needs and experiences.

# 10.Documenting Failed Experiences During the Study

## 10.1. Synthetic Dataset Creation Using Averaged Data

Initially, the research aimed to create a synthetic dataset by merging averaged life expectancy data from WHO [31] with other averaged indicators such as smoking rates, sleep duration, and BMI from various sources. The next step involved using the CTGAN model to generate individual-level data from these merged averages.

However, this approach faced significant challenges:

* Averaged Data Limitations: Since the data represented averages at the country level, it was ineffective in capturing meaningful individual relationships between indicators (e.g., smoking reducing life expectancy).
* Indicator Relationships: The approach also failed to detect nuanced interactions between multiple indicators (such as smoking and BMI) due to the averaged nature of the data.
* Dataset Insufficiency: Building a comprehensive synthetic dataset required collecting numerous datasets for each indicator, making the approach impractical at this stage.
* Poor Initial Results: Early attempts, using limited datasets (only four indicators), resulted in poor predictions. There was little observable relationship between indicators and life expectancy beyond the inherent country-level averages.
* Introduction of Penalties: The introduction of penalty scores showed slight improvement but failed to produce robust relationships and realistic predictive capabilities. The results remained heavily biased toward averaged data points.
* Poor Model Performance: Experiments with multiple machine learning models consistently produced unsatisfactory results, with predictions limited to narrow ranges of life expectancy, as evidenced by visualization graphs.

Due to these compounded issues, the decision was made to abandon this synthetic averaged data approach and identify more suitable, realistic datasets for future analyses.

**You can review our implementation on GitHub at the following link:** [**https://github.com/mahersalman/life-expectancy/tree/main/Archive**](https://github.com/mahersalman/life-expectancy/tree/main/Archive)

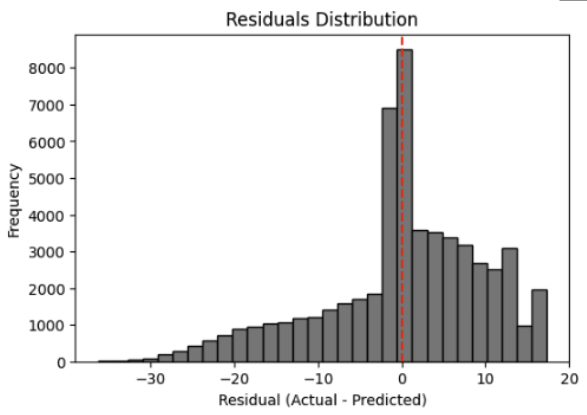
## 

## 10.2. Challenges Encountered in Deep Learning Approach

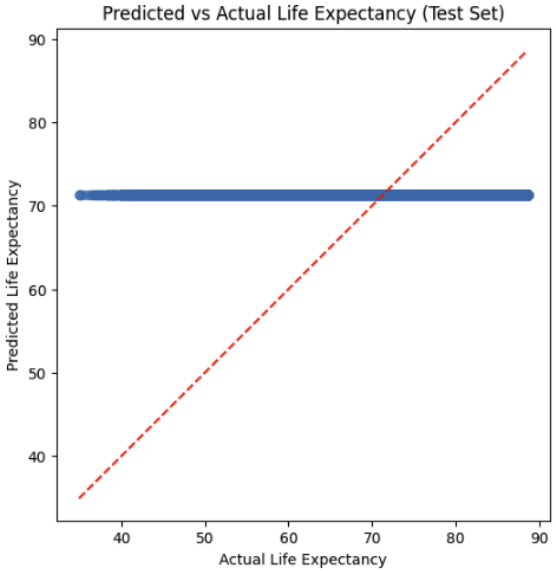
In an effort to leverage state-of-the-art architectures for tabular forecasting, we experimented with **SAINT** [21]. Our workflow included extensive hyperparameter sweeps (varying attention heads, embedding dimensions, learning rates, etc.) and protracted training runs—up to 20 epochs over several hours—yet the model consistently struggled to learn meaningful patterns from our synthetic life-expectancy dataset.

The residuals distribution (**Fig. 17**) reveals a pronounced bias and heavy tails, indicating that SAINT’s errors are both large and skewed. Likewise, the Predicted vs. Actual scatter plot (**Fig. 18**) shows most points collapsed around ~71 years rather than following the full 45–90 year range, demonstrating a failure to capture variance across the test set. Finally, the epoch-by-epoch training MSE log (**Fig. 19**) illustrates that after an initial drop, mean squared error plateaus rapidly, with minimal improvement even after 20 epochs.

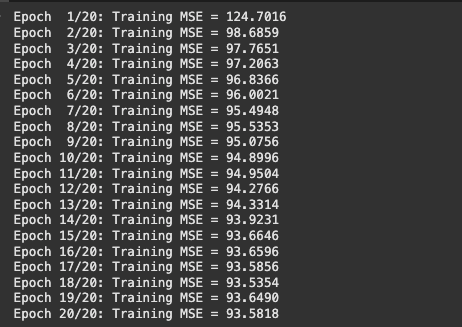
Despite repeated tuning, SAINT’s predictions remained plagued by high MSE and RMSE, failing to approximate the true variance in our labels. These negative results underscore that, for our particular tabular regime, conventional tree-based methods (e.g., XGBoost) or more specialized tabular-learning approaches may offer superior performance.



**Fig. 17 - histogram of (Actual – Predicted) showing a pronounced bias and heavy tails.**



**Fig. 18 - scatter plot on the test set, with most predictions collapsed around 71 years.**

****

**Fig. 19 - console output of mean squared error per epoch, illustrating minimal improvement after the first few rounds.**

# 

# References:

[1] Lipesa, B. A., Okango, E., Omolo, B. O., & Omondi, E. O. (2023). An application of a supervised machine learning model for predicting life expectancy. *SN Applied Sciences, 5*, Article 189. <https://doi.org/10.1007/s42452-023-05404-w>

[2] Malanin, V. (2025). Machine Learning Algorithms in Personalized Life Expectancy Prediction: A Systematic Review. International Journal of Science, Engineering and Management (IJSEM), 12(05), 67-74. <https://ijsem.org/article/11%20May%202025%20IJSEM.pdf>

[3] Stokanović, S., Đukić, D., & Miljković, N. (2023). Robustness of XGBoost algorithm to missing features for binary classification of medical data. *IEEE Xplore.* <https://ieeexplore.ieee.org/document/10495929>

[4] Singh, H., Mhasawade, V., Chunara, R. (2021). *Generalizability Challenges of Mortality Risk Prediction Models: A Retrospective Analysis on a Multi-center Database*. medRxiv. <https://www.medrxiv.org/content/10.1101/2021.07.14.21260493v1>

[5] Das, A., Uddin, M. M., & Karim, M. R. (2025). Predicting Life Expectancy using Machine Learning Techniques. International Journal of Statistical Sciences, 25(1), 55-70.

<https://www.researchgate.net/publication/390881873_Predicting_Life_Expectancy_using_Machine_Learning_Techniques>

[6] Georgiev, V., Hadzhikoleva, S., & Hadzhikolev, E. (2024). Impact of global country indicators on life expectancy. *Contemporary Scientific and Industrial Research, 1*(1), Article 11. <https://doi.org/10.70862/CSIR.2024.0101-04>

[7] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. <https://dl.acm.org/doi/pdf/10.1145/2939672.2939785>

[8] Streppel, M. T., Boshuizen, H. C., Ocké, M. C., Kok, F. J., & Kromhout, D. (2007). Mortality and life expectancy in relation to long‐term cigarette, cigar and pipe smoking: The Zutphen Study. Tobacco Control, 16(2), 107–113. <https://pmc.ncbi.nlm.nih.gov/articles/PMC2598467/>

[9] Reimers, C. D., Knapp, G., & Reimers, A. K. (2012). Does Physical Activity Increase Life Expectancy? A Review of the Literature. Journal of Aging Research, 2012, Article ID 243958. <https://pmc.ncbi.nlm.nih.gov/articles/PMC3395188/>

[10] Stenholm, S., Head, J., Kivimäki, M., Magnusson Hanson, L. L., Pentti, J., Rod, N. H., Clark, A. J., Oksanen, T., Westerlund, H., & Vahtera, J. (2019). Sleep Duration and Sleep Disturbances as Predictors of Healthy and Chronic Disease‐Free Life Expectancy Between Ages 50 and 75: A Pooled Analysis of Three Cohorts. \*J Gerontol A Biol Sci Med Sci\*, \*74\*(2), 204–210. <https://pubmed.ncbi.nlm.nih.gov/29415200/>

[11] Korhonen, K., Moustgaard, H., Tarkiainen, L., Ostergren, O., Costa, G., Urhoj, S. K., & Martikainen, P. (2021). Contributions of specific causes of death by age to the shorter life expectancy in depression: A register-based observational study from Denmark, Finland, Sweden and Italy. Journal of Affective Disorders, 295, 831–838. <https://www.sciencedirect.com/science/article/pii/S0165032721008818>

[12] Turin, T. C., Tonelli, M., Manns, B. J., Ravani, P., Ahmed, S. B., & Hemmelgarn, B. R. (2012). Chronic kidney disease and life expectancy. Nephrology Dialysis Transplantation, 27(8), 3182–3186. <https://pubmed.ncbi.nlm.nih.gov/22442392/>

[13] Majer, I. M., Nusselder, W. J., Mackenbach, J. P., Klijs, B., & van Baal, P. H. M. (2011). Mortality risk associated with disability: A population-based record linkage study. \*American Journal of Public Health, 101\*(12), e9–e15. <https://pmc.ncbi.nlm.nih.gov/articles/PMC3222426/>

[14] Liu, Y.-T., Lee, J. H., Tsai, M. K., Wei, J. C.-C., Wen, C.-P., & et al. (2022). The effects of modest drinking on life expectancy and mortality risks: a population-based cohort study. Scientific Reports, 12, Article 7476. <https://www.nature.com/articles/s41598-022-11427-x>

[15] Seshasai, S. R. K., Kaptoge, S., Thompson, A., Di Angelantonio, E., Gao, P., Sarwar, N., Paul, D. S., Manson, J. E., Sattar, N., Kelishadi, R., Engström, G., Kandala, N. B., Mohan, V., Ratonwala, S., Patel, R. A., Gillum, R. F., Lewington, S., & Danesh, J.; Emerging Risk Factors Collaboration. (2011). Diabetes mellitus, fasting glucose, and risk of cause‐specific death. \*New England Journal of Medicine\*, \*364\*(9), 829–841. <https://pmc.ncbi.nlm.nih.gov/articles/PMC4109980/>

[16] Shavelle, R. M., Paculdo, D. R., Kush, S. J., Mannino, D. M., & Strauss, D. J. (2009). Life expectancy and years of life lost in chronic obstructive pulmonary disease: findings from the NHANES III Follow-up Study. International Journal of Chronic Obstructive Pulmonary Disease, 4, 137–148. <https://pmc.ncbi.nlm.nih.gov/articles/PMC2672796/>

[17] American Heart Association News. (2019, July 25). Here's how many years you could gain by keeping heart disease at bay. American Heart Association. <https://www.heart.org/en/news/2019/07/25/heres-how-many-years-you-could-gain-by-keeping-heart-disease-at-bay>

[18] Vandergriendt, C. (2024, March 12). Stroke Prognosis: Factors Affecting Life Expectancy and Outlook. Healthline. <https://www.healthline.com/health/stroke/stroke-prognosis#life-expectancy>

[19] Lardieri, A. (2018, October 31). Study: High and low BMI linked to increased risk of death from all major causes. U.S. News & World Report. <https://www.usnews.com/news/health-care-news/articles/2018-10-31/study-high-and-low-bmi-linked-to-increased-risk-of-death-from-all-major-causes>

[20] Heywood, A. L. (2023, May 4). How many people have asthma? Healthline. Medically reviewed by Avi Varma, MD, MPH, AAHIVS, FAAFP. <https://www.healthline.com/health/asthma/how-many-people-have-asthma#life-expectancy>

[21] SAINT: Improved neural networks for tabular data via row attention and contrastive pre-training. *arXiv.* <https://arxiv.org/pdf/2106.01342>

[22]<https://xgboost.readthedocs.io/en/stable/install.html>

[23] WHO & The Lancet, 2024

<https://www.who.int/news/item/24-04-2024-global-immunization-efforts-have-saved-at-least-154-million-lives-over-the-past-50-years>

[24] WebMD. (2024, January). RA May Affect Your Life Expectancy. WebMD.<https://www.webmd.com/rheumatoid-arthritis/ra-life-expectancy>

[25] <https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease>

[26] <https://www.kaggle.com/code/alexisbcook/xgboost>

[27] <https://nextjs.org/>

[28] <https://tailwindcss.com/>

[29] <https://react.dev/>

[30] <https://www.typescriptlang.org/>

[31] [Life Expectancy (WHO). Kaggle](https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who)

[32]<https://flask.palletsprojects.com/en/stable/>

[33]<https://www.cdc.gov/nchs/fastats/life-expectancy.htm>

[34] <https://www.kaggle.com/datasets/joebeachcapital/life-expectancy?select=xmart.csv>

[35]<https://medium.com/@senihberkay/boosting-methods-57b74300235b>